

CENTRO UNIVERSITÁRIO DE ANÁPOLIS - UniEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

CAIRO MATEUS NEVES RIBEIRO

**Classificação do Uso e Cobertura do Solo do Estado
de Goiás Empregando Redes Neurais Artificiais**

ANÁPOLIS - GO

2019

CAIRO MATEUS NEVES RIBEIRO

**Classificação do Uso e Cobertura do Solo do Estado de Goiás
Empregando Redes Neurais Artificiais**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a conclusão do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Prof.^a M.^a Natasha Sophie Pereira

ANÁPOLIS - GO

2019

CAIRO MATEUS NEVES RIBEIRO

**CLASSIFICAÇÃO DO USO E COBERTURA DO SOLO DO ESTADO
DE GOIÁS EMPREGANDO REDES NEURAS ARTIFICIAIS**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 10 de dezembro de 2019, composta por:



Natasha Sophie Pereira
Presidente da Banca



Alison Carlos Filgueiras
Prof(a). Convidado(a)



Clarimar José Coelho
Prof(a). Convidado(a)

AGRADECIMENTOS

Um agradecimento para os amigos da faculdade que me ajudaram na caminhada da graduação, especialmente à Rodrigo Martins, Ketlen Fernandes, Sidney Junio, Eber Lucas, Rodrigo dos Santos, Davi Rezende, Luiz Henrique e tantos outros que não foram listados aqui.

Um agradecimento especial também a minha orientadora Natasha Sophie Pereira pela ajuda no processo de contextualização do problema, pelo suporte nos momentos de desespero e pelas suas valiosas correções.

Por fim, um agradecimento ao apoio que recebi dos meus pais, Valmir Gonzaga Ribeiro e Luzilívia Cláudia Neves Sena Pequeno Ribeiro, e de meu irmão, Cauwan Luiz Neves Ribeiro, que sempre apoiaram minhas escolhas e me incentivaram a continuar em frente.

*“Nem tudo o que reluz é ouro,
Nem todos os que vagueiam estão perdidos;
O velho que é forte não murcha,
Raízes profundas não são atingidas pela geada.”
(Senhor dos Anéis, Sociedade do Anel)*

RESUMO

Os dados de uso e cobertura do solo são ferramentas importantes para a fiscalização e o cumprimento da legislação em determinada área. Técnicas de redes neurais artificiais já foram empregadas com sucesso para na classificação de imagens de uso e cobertura do solo. Existem diferentes organizações que regulam normas para a classificação da distribuição do uso e cobertura do solo, como o IBGE. Esse trabalho buscou desenvolver um protótipo utilizando redes neurais artificiais que realiza a classificação do uso e cobertura do solo segundo as diretrizes do IBGE. Foi utilizada uma arquitetura de rede neural convolucional para classificação de imagens, essa rede foi treinada e possui três camadas convolucionais intercalada por três camadas de agrupamentos que realizaram o treinamento a partir de imagens de sensoriamento remoto obtidas do conjunto de dados *BigEarthNet*. Definida a rede e realizados o treinamento e a validação, a etapa de teste ocorreu com um conjunto de imagens referente ao estado de Goiás. Ao final desse trabalho verificou-se que a rede neural convolucional teve problemas em identificar as classes do IBGE para o conjunto de testes com dados de Goiás.

Palavras-chave: Redes Neurais Artificiais. Redes Neurais Convolucionais. Sensoriamento Remoto. Classificação e uso da cobertura do solo.

ABSTRACT

Land use and land cover data are important tools for monitoring and enforcing legislation in a given area. Artificial neural network techniques have been successfully employed for the classification of ground cover images. There are different organizations that regulate norms for the classification of land use distribution and land cover, such as IBGE. This work developed a prototype using artificial neural networks that classifies land use and land cover according to IBGE guidelines. A convolutional neural network architecture was used for image classification, this network was trained and has three convolutional layers interspersed by three layers of clusters that performed training from remote sensing images obtained from the BigEarthNet data set. Once the network was defined and the training and validation were performed, the test step was performed with a set of images related to the state of Goiás. At the end of this work, it was found that the convolutional neural network had problems identifying the IBGE classes for the set of images. tests with data from Goiás.

Keywords: Artificial Neural Networks. Convolution Neural Network. Remote Sensing. Classification and use of soil cover.

LISTA DE ILUSTRAÇÕES

Figura 1 – O neurônio biológico	22
Figura 2 – Conexão entre os neurônios de uma CNN	24
Figura 3 – Estrutura básica de uma rede neural convolucional	25
Figura 4 – Organização dos diretórios do BigEarthNet	34
Figura 5 – Arquivos disponibilizados pelo BigEarthNet	34
Figura 6 – Linhas de importação das bibliotecas	35
Figura 7 – Criação do mosaico e do arquivo TIF	35
Figura 8 – Conversão do TIF para JPG	36
Figura 9 – Exemplo de imagens geradas na conversão	36
Figura 10 – <i>Geotiff</i> utilizado para montar o conjunto de teste	37
Figura 11 – Classificação da Cobertura e Uso do Solo do estado de Goiás	38
Figura 12 – Sobreposição dos polígonos no <i>shapefile</i>	39
Figura 13 – Exemplo de imagens do conjunto de teste	39
Figura 14 – Resultados do treinamento e da validação para o <i>recall</i> e <i>F1 score</i>	42
Figura 15 – Função de perda no treinamento e validação	42
Figura 16 – Semelhança entre as classes	43

LISTA DE TABELAS

Tabela 1 – Tabela de formatos de imagens	17
Tabela 2 – Tabela de conversão entre as classes CLC e classes do IBGE	30
Tabela 3 – Tabela de bandas espectrais do <i>Sentinel-2</i>	32
Tabela 4 – Tabela de bandas espectrais do Landsat-8	32
Tabela 5 – Resultado para cada época do treinamento	41
Tabela 6 – Resultado para cada época da validação	41

LISTA DE ABREVIATURAS E SIGLAS

BIL	Bandas intervaladas por linhas - <i>Band Interleaved by Line</i>
BIP	Bandas intervaladas por pixels - <i>Band Interleaved by Pixel</i>
BSQ	Bandas sequenciais - <i>Band SeQuencial</i>
CNN	Rede Neural Convulucional - <i>Convolution Neural Network</i>
CUDA	Arquitetura de Dispositivo de Computação Unificada - <i>Compute Unified Device Architecture</i>
GPU	Unidade de processamento gráfico - <i>Graphics Processing Unit</i>
TP	Verdadeiro Positivo - <i>True Positive</i>
TN	Verdadeiro Negativo - <i>True Negative</i>
FP	Falso Positivo - <i>False Positive</i>
FN	Falso Negativo - <i>False Negative</i>
FOV	Campo de Visão - <i>Field of View</i>
GPS	Sistema de Posicionamento Global - <i>Global Positioning System</i>
IA	Inteligência Artificial - <i>Artificial Intelligence</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IFOV	Campo de Visão Instantânea - <i>Instantaneous Field of View</i>
ILSVRC	O Desafio de Reconhecimento Visual de Grande Escala do ImageNet - <i>The ImageNet Large Scale Visual Recognition Challenge</i>
K	Kelvin
ML	Aprendizado de Máquina - <i>Machine Learning</i>
ReLu	Unidade Linear Retificada - <i>Rectified Linear Unit</i>
REM	Radiação Eletromagnética
RNA	Redes Neurais Artificiais - <i>Artificial Neural Networks</i>
SIG	Sistema de Informação Geográfico

SR Sensoriamento Remoto - *Remote Sensing*

SWIR Imagens Infravermelhas de Ondas Curtas - *Short-wave Infrared Imagery*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Justificativa	13
1.2	Problema	13
1.3	Objetivos	14
1.3.1	Objetivos Gerais	14
1.3.2	Objetivos Específicos	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Imagens de Satélite	15
2.2	Inteligência Artificial	16
2.2.1	Aprendizado de Máquina	19
2.2.2	Redes Neurais Artificiais	21
2.3	Aprendizagem Profunda	23
2.3.1	Redes Neurais Convolucionais	24
2.3.2	Métricas de Desempenho	26
2.4	Tecnologia	28
3	MATERIAIS E MÉTODOS	29
3.1	Materiais	29
3.1.1	Área de Estudo	29
3.1.2	Bases de Dados	29
3.1.3	Imagens	31
3.2	Métodos	32
3.2.1	Pré-Processamento	32
3.2.2	Treinamento	33
3.2.3	Testes	33
4	DESENVOLVIMENTO	34
4.1	Imagens de treinamento e validação	34
4.2	Imagens de teste	37
4.3	Treinamento da Rede Neural Artificial	39
4.3.1	Resultados do Treinamento e validação	40
4.3.2	Resultados do Teste	43
5	CONSIDERAÇÕES FINAIS	44
	REFERÊNCIAS	45

ANEXOS

48

ANEXO A – CLASSES DE COBERTURA E USO DA TERRA - IBGE

49

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

A classificação do uso e cobertura do solo é realizada com a ajuda de Sistemas de Informações Geográficas (SIG). Esses sistemas abstraem a complexidade de desenhar os mapas a mão, porém, o processo para essa classificação ainda é realizado de forma manual.

A inteligência artificial, especialmente as técnicas de Aprendizado de Máquina (*Machine Learning*) se propõe a resolver problemas do tipo classificação e regressão. Utilizando algoritmos de classificação juntamente com técnicas de reconhecimento de padrões é possível encontrar soluções para os mais diferentes contextos, por exemplo: a classificação de microalgas (CORRÊA et al., 2017), a classificação de gêneros musicais brasileiros a partir de letras e acordes (JUNIOR, 2017), entre outros. Um subconjunto do Aprendizado de Máquina chamado de Redes Neurais Artificiais (RNA) já foi utilizado para realizar a classificação de problemas semelhantes à proposta deste trabalho, como pode ser visto no trabalho de Amato et al. (2015), intitulado *Remotely Sensed Soil Analysis Using Artificial Neural Networks: A Case Study of El-Fayoum Depression, Egypt*, que apresenta a classificação da cobertura e uso do solo utilizando como estudo de caso a Depressão de El-Fayoum no Egito e no trabalho de Hui, Wiele e Siamak (2009), intitulado *An Automated Artificial Neural Network System for Land Use/Land Cover Classification from Landsat TM Imagery*, que verificou a classificação da bacia do Rio Neuse na Carolina do Norte.

A partir disso, este trabalho classificou a cobertura e uso do solo segundo as diretrizes do Instituto Brasileiro de Geografia e Estatísticas (IBGE, 2018). A cada dois anos, o IBGE realiza levantamentos sobre a cobertura e uso do solo no Brasil, espacializando e quantificando este uso, o que possibilita a verificação da acurácia da rede neural a ser desenvolvida.

1.2 PROBLEMA

Para classificação de cobertura e uso do solo, são empregadas técnicas de mapeamento digital a partir do uso de Sistemas de Informações Geográficas (SIG). Além disso, existem diferentes instituições que definem diferentes tipos de classificação, tais como: Instituto Brasileiro de Geografia e Estatística (IBGE, 2019) no Brasil, *CORINE Land Cover* (CLC, 1995) na Europa, *United States Geological Survey* (USGS, 2019) nos Estados Unidos, entre outras. A presença de diferentes instituições, cada qual estabelecendo um conjunto de classificação de padrões, dificulta a adoção de uma classificação global. A inteligência artificial, mais especificamente as redes neurais artificiais, poderia ser utilizada para automatizar essa classificação dos padrões do IBGE para o cenário brasileiro. Dessa forma, como utilizar redes neurais artificiais para

classificação da cobertura e uso do solo no estado de Goiás seguindo as diretrizes do IBGE?

1.3 OBJETIVOS

1.3.1 OBJETIVOS GERAIS

Desenvolver um protótipo utilizando redes neurais artificiais para classificar a cobertura e uso do solo no estado de Goiás.

1.3.2 OBJETIVOS ESPECÍFICOS

- Discutir como é feita a aquisição de dados de satélites e como a técnica de rede neural artificial escolhida atua;
- Desenvolver a rede neural artificial;
- Treinar a rede neural artificial;
- Validar a técnica de rede neural artificial escolhida para o caso em estudo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 IMAGENS DE SATÉLITE

Os dados obtidos a partir de sensores imageadores, são imagens, que são utilizados como fonte de dados para realizar a classificação do uso e cobertura do solo. Gonzalez e Woods (2010, p.1), definem imagem como sendo: "[...] uma função bidimensional, $f(x, y)$, em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem nesse ponto".

Uma definição complementar que define a expressão matemática para a obtenção de valores para x e y é dada por Filho e Neto (1999, p.19), que afirma que:

a função $f(x, y)$ representa o produto da interação entre iluminância $i(x, y)$ - que exprime a quantidade de luz que incide sobre o objeto - e as propriedades de refletância ou de transmitância própria do objeto, que pode ser representado pela função $r(x, y)$.

Como pode ser visto uma imagem é qualquer função $f(x, y)$ que está em um plano e que possui pontos que recebem valores de intensidade ou de nível de cinza. Também é possível definir a função 2.1, que expressa o valor de cada ponto.

$$f(x, y) = i(i, x) * r(x, y) \quad (2.1)$$

Já imagem digital é uma representação discreta de uma imagem onde x , y e a intensidade possuem valores finitos, dessa forma uma imagem é composta por um número finito de elementos, cada um com valor e localização específicos (GONZALEZ; WOODS, 2010; SOLOMON; BRECON, 2013).

As imagens digitais podem ser divididas em duas categorias principais: vetoriais e *raster*. As imagens vetoriais são formadas usando equações matemáticas como linhas, pontos, polígonos e outros. A realização de mudanças nesse tipo de imagem é relativamente simples, como elas são criadas a partir de funções, basta alterar o valor das variável de entrada para mudar a imagem resultante. Esse tipo de formato normalmente é usado por designers gráficos. Já as imagens *raster* são imagens representadas com a utilização de pixels. Normalmente utilizadas em fotografias (MONTABONE, 2010, p.31).

Em sensoriamento remoto a varredura dos sensores é feita no terreno em linhas sequenciais, cada linha sendo amostrada por uma área unitária (pixel). Os pixels são registrados em uma grade quadrangular ou retangular e o valor do pixel corresponde ao valor da radiância, emitância ou retroespalhamento. No caso de sensoriamento remoto, o primeiro pixel da imagem situa-se

no canto superior esquerdo, assim as linhas são contadas de cima para baixo e as colunas da esquerda para a direita. O tamanho da imagem é dado pela multiplicação da quantidade de linhas pela quantidade de colunas e pelo número de bandas da imagem (MENESES; ALMEIDA, 2012, p.77-79).

Conforme os autores, pode-se observar que o pixel é um elemento crucial para a formação de imagens digitais *raster*. Os sensores de varredura capturam os elementos do terreno e os converte para imagens, sendo um único elemento com um valor e uma localização específica um pixel. Quanto ao formato de armazenamento das imagens de sensoriamento remoto, Meneses e Almeida (2012) citam, a existência de três formatos básicos, para armazenar fluxos de *bytes*:

- Bandas em sequência (BSQ, *Band SeQuencial Queue*): o formato mais simples, onde cada linha de dados é seguida imediatamente pela próxima linha na mesma banda e em seguida para a próxima banda da mesma imagem. Esse formato é ideal para quando se está trabalhando com bandas individuais e grandes áreas;
- Bandas intercaladas por linhas (BIL, *Band Interleaved by Line*): armazena a primeira linha da primeira banda, seguido pela primeira linha da segunda banda, seguindo assim e fazendo de forma intervalada pelo número de banda, ao final, reinicia-se o processo a partir da segunda linha da primeira, ao final, reinicia-se o processo a partir da segunda linha da primeira banda. Tipo preferível para processamentos mais pesados, como transformação de imagens digitais para imagens de reflectância; e
- Bandas intercaladas por *pixels* (BIP, *Band Interleaved by Pixel*): armazena o primeiro pixel de todas as bandas em ordem sequencial, seguida pelo segundo pixel para todas as bandas, e assim por diante, intervalada entre o número de *pixels*. Esse caso é preferível para a manipulação de todas as bandas de uma área pequena.

A necessidade de se criar métodos eficientes para exibição, armazenamentos e transmissão de imagens por rede, levou ao desenvolvimento de formatos padrões de imagens digitais. Os formatos de imagens consistem em um cabeçalho de arquivo e os valores numéricos de cada pixel (SOLOMON; BRECON, 2013). A tabela 1 expressa os formatos de imagens mais comuns.

2.2 INTELIGÊNCIA ARTIFICIAL

Segundo Coppin (2013, p.4) Inteligência Artificial (IA) "envolve utilizar métodos baseados no comportamento inteligente de humanos e outros animais para solucionar problemas complexos". Já para Elaine, Knight e Nair (2009), IA é "[...] o estudo de como computadores podem fazer tarefas de uma forma melhor do que as pessoas atualmente fazem". Ao se analisar essas definições, é possível inferir que Inteligência Artificial simula o comportamento inteligente humano em sistemas computacionais a fim de solucionar problemas de uma forma mais eficiente e, algumas vezes, mais eficaz, que os próprios humanos.

Tabela 1 – Tabela de formatos de imagens

Acrônimo	Denominação	Propriedades
GIF	<i>Graphics interchange format</i> [Formato para intercâmbio de imagens]	Limitado a apenas 256 cores (8 bits). compressão sem perda
JPEG	<i>Joint Photographic Experts Group</i> [Grupo Misto de Especialistas Fotográficos]	Formato em uso mais comum atualmente; compressão com perda; existe variante sem perda
BMP	<i>Bit map picture</i> [Imagem em mapa de bit]	Formato básico de imagem. conversão sem perda (em geral) limitada; existem variantes com perda
PNG	<i>Portable network graphics</i> [Imagens portáteis para redes]	Novo formato de compressão sem perda; projetado para substituir o GIF
TIF/TIFF	<i>Tagged image (file) format</i> [Formato (de arquivo) de imagem com rótulo]	Formato muito flexível, detalhado e adaptável; existem variantes comprimidas/descomprimidas

Fonte: Solomon e Brecon (2013)

O campo da inteligência artificial não tenta apenas compreender, mais também construir entidades inteligentes. Para o estudo da IA, e conseqüentemente a construção dessas entidades inteligentes, são definidas quatro diferentes abordagens cada uma apresentando métodos específicos, são elas: pensando como um humano, pensando racionalmente, agindo racionalmente e agindo como seres humanos (RUSSELL; NORVIG, 2013). Russell e Norvig (2013, p.4) apresentam definições interessantes para cada uma delas:

- Pensando como um humano: para determinar como programas pensam como seres humanos, é necessário definir como um ser humano pensa. É preciso penetrar nos componentes reais da mente humana. Existem três formas de fazer isso, através de introspecção, através de experimentos psicológicos e através de imagens cerebrais. Depois de obter uma teoria para a mente suficientemente precisa, será necessário expressar a teoria para um programa de computador;
- Pensando racionalmente: o uso do campo da lógica para resolução de problemas. A tradição logicista dentro da IA espera desenvolver programas que consigam resolver *qualquer* problema solucionável descrito por notação lógica para criar sistemas inteligentes. Existem dois problemas nessa abordagem, o primeiro é que não é fácil enunciar o conhecimento informal nos termos formais exigidos pela notação lógica, o segundo, são os recursos computacionais, já que problemas com alguns centenas de fatos podem esgotar esses recursos, a menos que ele tenha alguma orientação sobre as etapas de raciocínio que deve tentar primeiro;
- Agindo racionalmente: um agente simplesmente é aquele que age. Certamente todos os programas de computador realizam alguma coisa, mais espera-se que um agente computacional faça mais: opere sob controle autônomo, perceba seu ambiente, persista por um período de tempo prolongado, adapte-se a mudanças e seja capaz de criar e perseguir

regras. Um agente racional é aquele que age para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado. Na abordagem de leis do pensamento em IA. A ênfase É sempre dada as inferências corretas;

- Agindo como seres humanos: a base para essa definição é o teste de Turing. Proposto por TURING (1950), o teste foi projetado para fornecer uma definição operacional satisfatória de inteligência. No teste de Turing, o computador passará no teste se um interrogador humano, depois de propor algumas perguntas por escrito, não conseguir distinguir se as repostas que recebe estão vindo de um computador ou de uma pessoa. Para que um computador seja aprovado no Teste de Turing, é necessário que ele tenha capacidade de:
 - i) processar linguagem natural de modo que permita uma comunicação bem sucedida em um idioma natural;
 - ii) ser capaz de representar conhecimento, armazenando o que sabe ou ouve;
 - iii) possuir raciocínio automatizado e utilizar as informações armazenadas sempre com a finalidade de responder perguntas e tirar novas conclusões; e
 - iv) conseguir aprender e se adaptar a novas circunstâncias, detectar e extrapolar padrões, ou seja, exercer o Aprendizado de Máquina.

O Teste de Turing não considera a interação física direta entre entrevistador e entrevistado, para não influenciar o processo de julgamento da inteligência. Porém, uma versão completa do teste, chamada de Teste de Turing Total, prevê transmissão de vídeo de modo que o interrogador avalie as habilidades de percepção do entrevistado, além disso, permite a troca de objetos físicos entre os envolvidos. Neste caso, para que um computador seja aprovado, ele ainda precisa dominar

- v) a visão computacional, para ter a capacidade de perceber e reconhecer objetos; e a
- vi) robótica, para manipular os objetos e se movimentar.

As seis habilidades testadas pelo Teste de Turing e Teste de Turing Total são as disciplinas que compõem a maior parte de IA.

O objetivo da IA é criar computadores que consigam, agir ou pensar como humanos. Utilizando-se de diferentes abordagens, como adquirir o conhecimento do cérebro humano e transmitir esse problema para um computador, ou através de preposições lógicas, ou por meio de inferências ou ainda por meio de técnicas segmentadas com foco em diferentes resoluções de problemas que juntas poderiam ser utilizadas para provar o teste de Turing. O campo da IA como dito por Russell e Norvig (2013) tem como objetivo construir entidades inteligentes.

2.2.1 APRENDIZADO DE MÁQUINA

Aprendizado de Máquina (AM) é o processo de indução de uma hipótese (ou aproximação de uma função) a partir da experiência passada para um computador. O AM surgiu a partir da criação de técnicas para reduzir a necessidade de intervenção humana, técnicas que permitiam ao computador criar suas próprias hipóteses ou funções a partir das experiências vivenciadas por ele, de modo que fosse capaz de resolver um problema proposto. O AM permite ao computador obter conclusões genéricas a partir de um conjunto específico de exemplos, de modo que as hipóteses são geradas a partir de dados representativos de instâncias do problema, que são conhecidos como Conjunto de Dados Faceli et al. (2011, p.2). Mitchell (1997) define o aprendizado de máquina como: "A capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência".

Aprendizado de máquina é a capacidade de se adaptar a novas circunstâncias, detectar e extrapolar padrões. Pode-se dizer que o aprendizado de uma função geral ou regra (possivelmente incorreta) a partir de pares específicos de entrada-saída é chamado de aprendizagem indutiva Russell e Norvig (2013).

O campo do aprendizado de máquina se concentra na aproximação de uma hipótese ou função a partir da experiência adquirida pelo computador. É possível observar também que o AM se utiliza de indução para obter essa hipótese. A experiência em AM é obtida a partir de dados do domínio do problema que se deseja resolver, esses dados são conhecido como conjunto de dados.

Existem três principais tipos de aprendizagem, conforme pode ser visto em Russell e Norvig (2013, p.606-607): aprendizagem não supervisionada, aprendizagem por reforço e aprendizagem supervisionada. Muitas vezes essas distinções nem sempre são tão nítidas, como pode ser visto na aprendizagem semissupervisionada.

- Aprendizagem não supervisionada: o agente aprende padrões na entrada, embora não seja fornecido nenhum *feedback* de rotulação dos valores. A tarefa mais comum no aprendizado não supervisionado é o agrupamento: a detecção de grupos de exemplos de entrada potencialmente úteis. Outras atividades realizadas são a associação e a sumarização;
- Aprendizagem por reforço: o agente aprende a partir de uma série de reforços, que consistem em recompensas e punições;
- Aprendizagem supervisionada: o agente observa os pares de entrada-saída de alguns exemplos fornecidos, e aprende uma função que faz o mapeamento da entrada e da saída. São principalmente utilizados para realizar classificação e regressão;
- Aprendizagem semissupervisionada: são dados alguns poucos exemplos rotulados e deve-se fazer que puder de uma grande coleção de exemplos não rotulados.

O principal desafio do aprendizado de máquina é um algoritmo que seja preciso para novas entradas, diferentes daquelas utilizadas na fase de treinamento. A capacidade de se adaptar a novos conjuntos de dados é chamada de generalização. Normalmente ao treinar um modelo de aprendizagem de máquina, é apresentado a rede um conjunto de treinamento, dessa forma é possível calcular o conjunto de valores incorretos nesta fase, que é chamado de erro de treinamento. Para separar o AM da otimização é necessário também realizar um cálculo do erro de generalização. O erro da generalização é definido como o valor esperado do erro em uma nova entrada. Geralmente o erro de generalização de um modelo de aprendizado de máquina é medido através de um conjunto de teste diferente do conjunto de treinamento utilizado (GOODFELLOW; BENGIO; COURVILLE, 2016).

Podem acontecer alguns problemas na generalização dos dados em AM. Quando o modelo oferece uma baixa capacidade de generalização, é dito que o algoritmo está superajustando os dados de treinamento (*overfitting*), nesse cenário a hipótese memorizou ou se especializou no conjunto de dados. Um outro problema que pode acontecer é a condição de subajustamento (*underfitting*), isso pode acontecer quando o algoritmo de AM apresenta uma baixa taxa de acerto mesmo para o subconjunto de treinamento, a rede não consegue representar todos os padrões de dados de saída esperados (FACELI et al., 2011).

Um dos grandes desafios do aprendizado de máquina é definir um algoritmo generalista, ou seja, um algoritmo que consiga se adaptar mesmo para entradas não informadas no conjunto de treinamento. Além disso, alguns problemas podem surgir no treinamento de um algoritmo de AM: *overfitting* e *underfitting*. O *overfitting* é quando a rede memoriza o conjunto de treinamento e não consegue ser generalista para outras situações. O *underfitting* surge quando o algoritmo de AM não consegue uma acuracidade alta para o conjunto de treinamento, dessa forma, não é possível treinar todos os conjuntos de dados.

Estima-se que a quantidade de dados armazenados em bancos do mundo dobre a cada 20 meses. Esses dados podem assumir diferentes formatos como: séries temporais, *itemsets*, transações, grafos ou redes sociais, textos, páginas web, imagens (vídeos) e áudios (músicas). Esses conjuntos de dados podem representar objetos físicos, como uma cadeira, ou objetos abstratos, como sintomas de uma doença de um paciente que vai a um hospital. Em geral cada objeto pode ser descrito como um conjunto de atributos ou um vetor de características. Formalmente esses dados objetos em uma matriz $X_{n \times d}$, onde n é o número de objetos e d o número de atributos de entrada de cada objeto (FACELI et al., 2011; WITTER; FRANK, 2005).

Os dados que são utilizados pelas técnicas de AM necessitam de uma etapa de pré-processamento, como pode ser visto em Faceli et al. (2011, p.11):

Apesar do crescente número de bases de dados disponíveis, na maioria das vezes não é possível utilizar algoritmos de AM diretamente sobre esses dados. Técnicas de pré-processamento são frequentemente utilizadas para tornar os conjuntos de dados mais adequados para o uso de algoritmos de AM. Essas técnicas podem ser agrupadas nos seguintes grupos de tarefas: eliminação manual

de atributos, integralização de dados, amostragem de dados, balanceamento de dados, limpeza de dados, redução da dimensionalidade e transformação de dados.

Vários algoritmos de AM tem dificuldades de trabalhar com dados inconsistentes, redundantes ou que precisem lidar com um número muito grande de dados, essa quantidade pode estar relacionada a fatores como a quantidade de objetos, o número de atributos ou ambos. Em dados reais a distribuição dos objetos entre classes pode não ser uniforme, formando um desbalanceamento do conjunto de dados, alguns algoritmos de AM tem dificuldade em induzir um bom modelo a partir de conjuntos desbalanceados (FACELI et al., 2011). Em aplicações reais, normalmente são encontrados vários desses problemas relacionados aos dados, isso dificulta criar um modelo que se baseie em conjuntos de dados balanceados.

2.2.2 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais (RNA's) foram desenvolvidas tendo como inspiração o funcionamento do sistema nervoso humano, e tem o objetivo de simular a capacidade de aquisição de conhecimento do cérebro na aquisição de conhecimento (FACELI et al., 2011). (HAYKIN, 2001) apresenta a seguinte definição para RNA:

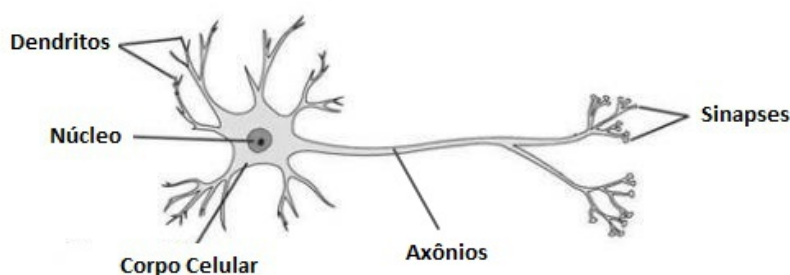
Uma rede neural é um processador maciçamente paralelamente distribuído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ele se assemelha ao cérebro em dois aspectos: o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem e as forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizada para armazenar o conhecimento adquirido.

As redes neurais artificiais tentam replicar a capacidade do sistema nervoso de armazenar e processar conhecimento. Elas são processadores que realizam operações de forma paralela e distribuída. Além disso, conseguem armazenar o conhecimento previamente adquirido entre as conexões, em forças de conexão entre os neurônios que são conhecido como pesos sinápticos.

O sistema nervoso, que faz parte do cérebro, é um conjunto de células que determinam o funcionamento e o comportamento dos seres vivos. Sua unidade fundamental é o neurônio, célula nervosa, que se distingue das outras células pela sua capacidade de apresentar excitabilidade, o que a permitindo responder a estímulos internos e externos. O neurônio biológico é o principal bloco de construção do cérebro, sendo ele dividido em: dendritos, corpo celular e axônio (FACELI et al., 2011, p.108). A Figura 1 apresenta um modelo de neurônio biológico.

Cada neurônio consiste em um corpo celular ou soma, que contém o núcleo celular. Ramificando o corpo existem uma série de fibras chamadas de dendritos e uma única fibra longa chamada axônio. UM axônio pode chegar a até 1 metro de comprimento. O neurônio faz conexões entre 10-100.000 outros neurônios, em junções chamadas sinapses (RUSSELL; NORVIG,

Figura 1 – O neurônio biológico



Fonte: Adaptado de Fred (2017)

2013). Os modelos biológicos de neurônios serviram como base para a criação dos neurônios artificiais que aplicados nas redes neurais artificiais.

O modelo para um neurônio que forma a base das redes neurais artificiais, apresenta três modelos básicos que se assemelham ao neurônio biológico: o conjunto de sinapses ou elos de conexão, cada um caracterizado por um peso e uma força própria, um somador que realizar as operações de soma para os sinais de entrada e uma função de ativação que restringe a amplitude da saída do neurônio. O modelo neural inclui também um *bias*, este por sua vez tem o efeito de aumentar ou diminuir a entrada líquida de uma função de ativação (HAYKIN, 2001, p.36-37).

Haykin (2001, p.37) apresenta as Equações (2.2) e (2.3) para descrever um neurônio k . Para a Equação (2.2): x_1, x_2, \dots, x_m são os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k ; u_k é a saída do combinador linear. Já a Equação (2.3), b_k é o bias, $\varphi(*)$ é a função de ativação e y_k é o sinal de saída do neurônio.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.2)$$

$$y_k = \varphi(u_k + b_k) \quad (2.3)$$

A informação em uma rede neural normalmente flui da camada de entrada para uma camada de saída, já para redes com muitas camadas esse fluxo acontece camada a camada. As conexões de retroalimentação permitem que o neurônio receba em seus terminais de entrada a saída da mesma camada ou de uma camada posterior. Redes sem conexão de retropropagação, normalmente utilizadas na prática, são denominadas RNAs *feedforward*. O número de camadas, o número de neurônios em cada camada, o grau de conectividade e a presença ou não de retropropagação definem a topologia de uma RNA (FACELI et al., 2011, p.112).

Faceli et al. (2011, p.113-120) apresenta algumas arquiteturas de RNA's, são elas: perceptron, adaline, perceptron multicamadas.

- Perceptron: o perceptron é a forma mais simples de uma rede neural usada para a classificação de padrões ditos linearmente separáveis. Basicamente, ele consiste de um único neurônio com pesos sinápticos ajustáveis e bias (HAYKIN, 2001, p.143);
- Adeline: rede neural de apenas uma camada que apresenta uma função de ativação linear, essa rede utiliza um ajuste chamado de Regra Delta, cujo nome vem do uso de diferentes valores de saída desejada e da saída produzida (FACELI et al., 2011);
- Perceptron multicamadas: normalmente a rede consiste de um conjunto de unidades sensoriais que constituem as camadas de entradas, uma ou mais camadas ocultas e uma camada de saída. Ela consegue resolver diversos problemas difíceis através de um treinamento supervisionado utilizando um algoritmo de retropropagação de erro (*error back-propagation*) (HAYKIN, 2001).

Uma dificuldade que se tinha no treinamento de redes multicamadas era a ausência de um algoritmo capaz de treinar a rede. O algoritmo de *back-propagation*, baseado no gradiente descendente surgiu para resolver esse problema, ele necessita de uma função de ativação contínua, diferenciável e, de preferência, não decrescente (FACELI et al., 2011).

2.3 APRENDIZAGEM PROFUNDA

Aprendizagem profunda (*Deep Learning*, DL) é uma sub-área da aprendizado de máquina, que emprega algoritmos para processar dados e imitar o processamento feito pelo cérebro humano (DATA SCIENCE ACADEMY, 2017).

Aprendizagem profunda e redes neurais profundas (*Deep Neural Networks*, DNNs) são redes neurais com múltiplas camadas ocultas. Um dos grandes fatores para a utilização da aprendizagem profunda não está vinculada a quantidades de camadas mais sim a capacidade de aprendizagem. Uma rede neural profunda não aprende simplesmente a prever uma saída para determinada entrada, ela compreende as características básicas das entradas. Outras redes neurais não possuem esse nível de abstração (VASILEV et al., 2019, p.68).

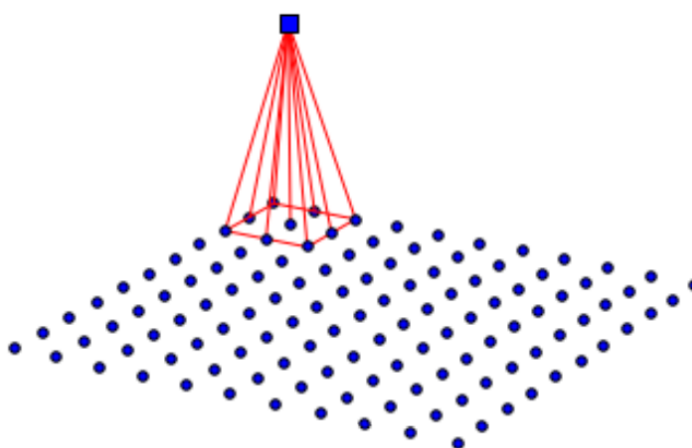
Pode-se definir aprendizagem profunda como uma das técnicas de aprendizado de máquina onde as informações são processadas em camadas hierárquicas e onde características dos dados aumentam a complexidade. Na prática todos os algoritmos de aprendizado profundo são algoritmos de aprendizado de máquina que compartilham algumas propriedades básicas comuns. Alguns algoritmos utilizados para aprendizado profundo, são: perceptron de múltiplas camadas, redes neurais convolucionais, redes recorrentes e *autoencoders* (VASILEV et al., 2019, p.73-74).

2.3.1 REDES NEURAIIS CONVOLUCIONAIS

Redes neurais convolucionais (CNNs) são um tipo especializado de redes neurais artificiais para processamento de dados que possuem uma topografia semelhante a de grade. O nome redes neurais convolucionais indica que a rede aplica uma operação matemática chamada de convolução. A convolução é um tipo especial de operação linear. As redes neurais convolucionais são redes neurais que em pelo menos uma de suas camadas utilizam a convolução ao invés da multiplicação geral da matriz (GOODFELLOW; BENGIO; COURVILLE, 2016, p.330).

Supondo que uma rede neural totalmente conectada tente tratar uma imagem de $32 * 32$ pixels, apenas para a camada de entrada são necessários 1024 neurônios. Assumindo que a primeira camada oculta tenha 2072 neurônios, temos um total de $1024 * 2072 = 2.121.728$ pesos. Essa quantidade de pesos gera um problema de memória necessária para realizar a operação. Outro problema apresentado para uma rede neural nessa situação é o *overfitting*. As CNNs possuem propriedades que auxiliam na resolução desses problemas, são elas: as conexões dos neurônios correspondem apenas aos vizinhos da imagem, dessa forma, os neurônios são forçados a receber informações de outros neurônios especialmente próximos isso reduz a quantidade de pesos já que nem todos os neurônios estão interconectados e elas utilizam compartilhamento de parâmetro (*parameter sharing*), entre outras palavras um número limitado de pesos são compartilhados entre todos os neurônios de uma camada (VASILEV et al., 2019, p.94). A Figura 2 apresenta essa conexão entre os neurônios vizinhos.

Figura 2 – Conexão entre os neurônios de uma CNN



Fonte: Vasilev et al. (2019, p.95)

Imagine o rastreamento de uma espaçonave com um sensor a laser, obtemos os valores para o tempo (t) e para a distância da espaçonave ($x(t)$). Agora suponha o laser apresente problema de ruídos nos dados obtidos e para diminuir esse ruído gostaríamos de calcular a média entre várias medidas. Normalmente as medições mais recentes são mais importantes por isso é feito uma média ponderada atribuindo maior valor aos dados mais recentes. Isso é feito com uma função de ponderação $w(a)$. Se aplicarmos essa operação a cada momento, obtemos uma função

de suavização que fornece uma estimativa suavizada da posição da espaçonave, essa operação é chamada de convolução. A operação de convolução normalmente é apresentada por um asterisco (GOODFELLOW; BENGIO; COURVILLE, 2016).

As Equações (2.4) e (2.5) representam as equações para uma convolução para a equação (2.5) e usando como base a terminologia de uma rede convolucional, o argumento x faz referência aos parâmetros de entrada, w faz referência ao kernel e a saída $s(t)$ faz referência ao mapa de recursos (*features map*) (GOODFELLOW; BENGIO; COURVILLE, 2016, p.331-332).

$$s(t) = \int x(a)w(t - a)da \quad (2.4)$$

$$s(t) = (x * w)(t) \quad (2.5)$$

Uma camada de agrupamento (*pooling*) divide os dados de entrada em uma grade, onde cada célula da grade representa um campo receptivo de vários neurônios. Em seguida uma operação de agrupamento é aplicada sobre cada elemento da grade. Existem diferentes tipos de camada de agrupamento e ela não altera a profundidade do volume por que a operação é executada de forma independente em cada fatia (VASILEV et al., 2019, p.108).

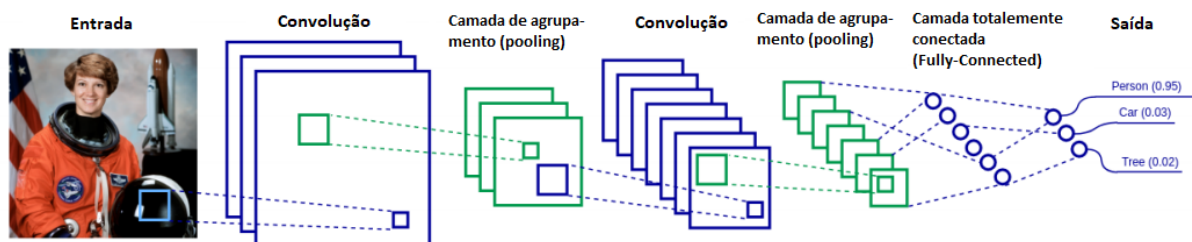
Existem as seguintes camadas de agrupamento: *max pooling*, *average pooling*, L^2 *pooling* e *weight average pooling* (VASILEV et al., 2019; GOODFELLOW; BENGIO; COURVILLE, 2016).

O tipo mais comum de função de ativação para as CNN é a ReLu (*Rectified Linear Unit*). A função 2.6 descreve o funcionamento da ReLu (VASILEV et al., 2019). Para valores inferiores a zero a função de ativação atribui o valor zero, enquanto que para valores maiores ou iguais a zero ela atribui o respectivo valor, dessa forma, podemos dizer que o alcance de valores possíveis para a função de ativação ReLu é de 0 até ∞ .

$$f(a) = \begin{cases} a & \text{se } a \geq 0 \\ 0 & \text{se } a < 0 \end{cases} \quad (2.6)$$

A Figura 3 apresenta a estrutura básica de uma rede neural convolucional.

Figura 3 – Estrutura básica de uma rede neural convolucional



Fonte: Adaptado de Vasilev et al. (2019, p.110)

As redes neurais convolucionais possuem algumas características básicas, normalmente são realizadas alterações de uma camada de agrupamento entre uma ou mais camadas convolucionais, dessa forma, as camadas convolucionais conseguem detectar características em todos os níveis do tamanho do campo receptivo, outra característica é que as camadas convolucionais são utilizadas para extrair dados de entrada, os recursos detectados pelas camadas mais ocultas são altamente abstratos, dessa forma, esses dados não são legíveis para humanos, para resolver esse problema são utilizadas camadas totalmente conectadas (*fully-connected*) após a última camada de convolução/agrupamento, por último as camadas convolucionais mais profundas geralmente possuem mais filtros do que as iniciais (VASILEV et al., 2019, p.110-111).

2.3.2 MÉTRICAS DE DESEMPENHO

Para realizar a medição da performance do algoritmo de classificação é necessário a realização de testes feitos a partir de um conjunto de testes. Deste conjunto temos o conhecimento prévio dos resultados esperados, ele é então submetido ao algoritmo treinado que retornará os resultados obtidos. A comparação entre os resultados obtidos e os resultados esperados gera valores que são utilizados para calcular o desempenho do modelo (KELLEHER; NAMEE; D'ARCY, 2015).

Para medir a precisão dos classificadores normalmente é utilizada uma matriz de confusão, uma ferramenta muito útil para analisar a eficiência do classificador para diferentes classes. Ela define quatro termos utilizados nos cálculos das métricas (HAN; KAMBER; PEI, 2018). Essas termos são:

- Verdadeiro positivo (*True Positive*, TP): são casos onde os resultados esperados são verdadeiros e foram classificados com o rótulo correto;
- Verdadeiro negativo (*True Negative*, TN): são casos onde os resultados esperados são negativos e foram classificados com o rótulo correto;
- Falso positivo (*False Positive*, FP): são casos onde os resultados esperados são negativos e foram classificados incorretamente como positivo;
- Falso negativo (*False Negative*, FN): são casos onde os resultados esperados são verdadeiros e foram classificados incorretamente como falso;

A matriz de confusão é uma tabela com dimensão $m \times m$, sendo m o número de classes de entrada. Para obter uma boa acurácia com o classificador é ideal que a maioria dos valores se encontre na diagonal principal e o restante das colunas apresenta valores próximos ou iguais a zero (HAN; KAMBER; PEI, 2018).

Para problemas de classificação com dados balanceados, onde as classes possuem a mesma quantidade de entradas, são aplicadas métricas como a acurácia, em classes desbalancea-

das pode ser usadas métricas como precisão e *recall* e para problemas de classificação de várias classes normalmente são empregadas técnicas de precisão média, como o *F1 Score* (CHOLLET, 2018).

A acurácia é definida pela porcentagem do conjunto de testes que é classificada corretamente. A Equação (2.7) representa a acurácia, ela pode ser definido também com a taxa geral de reconhecimento do classificador (HAN; KAMBER; PEI, 2018). Para a acurácia somamos os valores da diagonal principal da matriz de confusão e dividimos pelo valor total de elementos do conjunto de teste.

$$Acurácia = \frac{TP + TN}{Total} \quad (2.7)$$

A partir da acurácia é possível calcular a taxa de erro, a taxa de erro ou taxa de classificação incorreta é um cálculo onde é subtraído 1 da *Acurácia*(M). Ela pode ser calculada pela Equação (2.8) (HAN; KAMBER; PEI, 2018). Seu objetivo é o contrário da acurácia, ela tenta expressão a quantidade de erro apresentado pelo algoritmo de classificação.

$$ErrorRate = \frac{FP + FN}{Total} \quad (2.8)$$

A precisão pode ser considerada uma medida de exatidão, ela mede a quantidade de elementos verdadeiros que são realmente definidos como verdadeiros positivos (POWERS, 2015). Seu cálculo é realizado pela divisão dos verdadeiros positivos pela soma dos verdadeiros positivos mais os falsos positivos, conforme apresentado na Equação (2.9).

$$Precisão = \frac{TP}{TP + FP} \quad (2.9)$$

O *recall* é uma medida de completude, ele tem como característica definir a porcentagem de verdadeiros positivos em relação aos falsos negativos. Seu cálculo pode ser observado na Equação (2.10) (HAN; KAMBER; PEI, 2018). Nessa equação temos que os verdadeiros positivos são divididos pelos verdadeiros positivos mais os falsos negativos.

$$Recall = \frac{TP}{TP + FN} \quad (2.10)$$

O *F1 score* faz referência aos verdadeiros positivos e a média entre os positivos previstos e os positivo reais, sendo está uma taxa construída para normalizar os valores idealizados (HAN; KAMBER; PEI, 2018). O *F1 score* faz a união entre a precisão e o *recall*, ele tenta indicar a qualidade geral do modelo de classificação.

$$F1Score = \frac{2 * Precisão * Recall}{Precisão + Recall} \quad (2.11)$$

2.4 TECNOLOGIA

Existem várias bibliotecas de código aberto que permitem criar códigos de redes neurais profundas em *Python*, sem a necessidade de escrever código do zero, são exemplos populares o *TensorFlow*, *Keras* e *PyTorch* (VASILEV et al., 2019).

Essas bibliotecas apresentam algumas características em comum, são elas: a unidade básica de armazenamento chama-se tensor, redes neurais são representadas como operações computacionais em grafos, todas as bibliotecas oferecem diferenciação automática, todas as bibliotecas usam a linguagem de programação *Python* e todas oferecem suporte a arquitetura de dispositivo de computação unificada (CUDA, *Compute Unified Device Architecture*) para operações em unidades gráficas de processamento (GPU, *Graphic Processing Unit*) (VASILEV et al., 2019).

Python é uma linguagem que proporciona o fácil aprendizado e que possui uma sintaxe de fácil utilização. Além disso, ela é uma linguagem expressiva, isto é, escreve-se menos código do que linguagens como C++ e Java. Outra característica do *Python* ser uma linguagem multiplataforma, dessa forma, um código escrito em um computador com *Windows* executaria normalmente em outra plataforma, como o *Linux* (SUMMERFIELD, 2018).

O *TensorFlow* é a biblioteca mais popular de DL, ela é desenvolvida e mantida pela *Google*, além disso, ela realiza automaticamente o controle do uso de GPU's se a máquina apresentar esse recurso, se a máquina tiver mais de uma GPU é necessário especificar qual delas será utilizada. O *TensorFlow* tem uma curva de aprendizado acentuada em comparação com outras bibliotecas (VASILEV et al., 2019).

O *Keras* é uma biblioteca de rede neural de alto nível que roda em cima de bibliotecas como o *TensorFlow*, *Theano* e *CNTK*. Com o *Keras* podem ser realizadas experimentações de forma bastante rápida e relativamente fácil em relação ao *TensorFlow* (VASILEV et al., 2019).

3 MATERIAIS E MÉTODOS

3.1 MATERIAIS

3.1.1 ÁREA DE ESTUDO

O estado de Goiás possui uma área territorial de $340.125,715 \text{ km}^2$ com densidade demográfica de $17,65 \text{ hab/km}^2$, sendo o sétimo maior estado em extensão territorial do Brasil, e possui uma população estimada de 6.921.161 pessoas segundo o censo de 2018. Possui 246 municípios e faz limite com os estados do Mato Grosso do Sul, Mato Grosso, Tocantins, Bahia, Minas Gerais e Distrito Federal (IBGE, 2018).

O Produto Interno Bruto (PIB) referente ao estado corresponde a 181.692 milhões de reais, isso corresponde ao nono PIB dentre os estados brasileiros. Seu Índice de Desenvolvimento Humano (IDH) é igual a 0,735 o que corresponde ao oitavo estado no ranking de qualidade de vida (IBGE, 2018).

A agropecuária é uma das maiores forças econômicas do Brasil, colecionando resultados positivos em 2017 e servindo de pilar para o crescimento econômico nesse período. A agropecuária goiana teve um bom desempenho com elevação de 24,6% na produção de grãos, visto que o Estado é um grande produtor de soja e milho, além da cana-de-açúcar. Ainda, o Estado se destaca na criação de gado e aves para corte (FAEG, 2017).

3.1.2 BASES DE DADOS

Foi utilizada a base de dados de imagens disponibilizada pelo *BigEarthNet*. O *BigEarthNet* disponibiliza 590.326 diferentes classes em imagens retiradas do *Sentinel-2*, adquiridas entre junho de 2017 e maio de 2018 de dez diferentes países europeus (Áustria, Bélgica, Finlândia, Irlanda, Kosovo, Lituânia, Luxemburgo, Portugal, Sérvia e Suíça). Essas imagens foram tratadas com um algoritmo para correção atmosférica, chamado de *sen2cor*. A partir disso, foi feita uma divisão desse conjunto de imagens sem sobreposição e cada conjunto de imagem foi anotado pela sua multi-classe de cobertura de uso do solo de acordo com as classes fornecidas pelo *CORINE Land Cover* (CLC) referente ao ano de 2018 (SUMBUL et al., 2019).

O *BigEarthNet* utiliza a classificação de *Level 3* do CLC, que divide o uso e cobertura do solo em 44 classes, sendo elas: tecido urbano contínuo, tecido urbano descontínuo, unidades industriais ou comerciais, redes rodoviárias e ferroviárias e terras associadas, áreas portuárias, aeroportos, locais de extração mineral, locais de despejo, locais de construção, áreas verdes urbanas, esporte e lazer, áreas aradas não irrigadas, terra permanentemente irrigada, campos de arroz, vinhedos, árvores frutíferas, oliveiras, pastagem, culturas anuais associadas a culturas

permanentes, padrões de cultivo complexo, terras principalmente ocupadas pela agricultura com áreas significativas de vegetação natural, áreas agroflorestais, florestas de folhas largas, florestas de coníferas, floresta mista, pastagens naturais, muros e charnecas, vegetação esclerófila, arbustos da floresta de transição, praias, dunas e areia, pedras nuas, áreas escassamente vegetadas, áreas queimadas, geleiras e neves perpétuas, pântanos interiores, turfeiras, saleiras, salinas, planícies lamacentas, cursos d'água, corpos d'água, lagoas costeiras, estuários e mar e oceano (SUMBUL et al., 2019; CLC, 2006).

Para o estudo de caso será utilizada a classificação do uso e cobertura do solo fornecida pelo IBGE (2018), conforme disponibilizado no anexo A. Dessa forma, é necessário realizar uma conversão entre as multi-classes disponibilizadas no *BigEarthNet* e as 12 classes definidas pelo IBGE. Assim, foi realizada uma conversão entre as classes conforme estabelecido na Tabela 2.

Tabela 2 – Tabela de conversão entre as classes CLC e classes do IBGE

Classe do IBGE	Respectivas classes do CLC
Área Artificial	Tecido urbano contínuo, tecido urbano descontínuo, unidades industriais ou comerciais, redes rodoviárias e terras associadas, áreas portuárias, aeroportos, locais de extração mineral, locais de despejo, locais de construção, áreas verdes urbanas, esporte e lazer
Área Agrícola	Áreas aradas não irrigadas, terra permanentemente irrigada, campos de arroz, vinhedos, árvores frutíferas, oliveiras, padrões de cultivos complexos e culturas anuais associadas a culturas permanentes
Pastagem com Manejo	Pastagem e pastagens naturais
Mosaico de Ocupações em Área Florestal	Terras principalmente ocupadas pela agricultura com áreas significativas de vegetação natural
Silvicultura	-
Vegetação Florestal	Florestas de folhas largas, florestas de coníferas, floresta mista
Área Úmida	Muros e charnecas, pântanos interiores, turfeiras, planícies lamacentas
Vegetação Campestre	Vegetação esclerófila, arbustos da floresta de transição
Mosaico de Ocupações em Áreas Campestres	Áreas agroflorestais
Corpo d'água Continental	Cursos d'água, corpos d'água
Corpo d'água Costeiro	Lagoas costeira, estuários, salinas, saleiras, mar e oceano
Área Descoberta	Praias, dunas e areia, pedras nuas, áreas escassamente vegetadas

Fonte: Adaptado de IBGE (2018) e CLC (2006)

Foram desconsideradas as classes de geleiras e neves perpétuas e áreas queimadas, por não se adequarem as classes propostas pelo IBGE (2018). Além disso, não houve nenhuma classe que pôde ser convertida como silvicultura, dessa forma, ela ficou sem representação pelo conjunto de dados que será utilizado.

3.1.3 IMAGENS

As imagens que são utilizadas neste trabalho, foram obtidas a partir de sensores remotos, mais especificamente aqueles acoplados nos satélites: *Sentinel-2* e *Landsat-8*. Para entender os sensores remotos é necessário entender quatro características gerais, conforme é definido em Novo (2010) e Mather (2004), são elas: i) resolução espacial; ii) resolução espectral; iii) resolução radiométrica; iv) resolução temporal.

A resolução espacial se refere as minúncias que podem ser distinguidas na imagem, e representam a menor feição passível de detecção realizada pelo instrumento em questão. É possível definir duas medidas importantes, sendo elas, o campo de visão (*Field of View*, FOV) e o campo de visão instantâneo (*Instantaneous Field of View*, IFOV). O FOV se refere a largura da faixa imageada pelo sensor. Enquanto que o IFOV se refere a área da superfície terrestre que é focada por ele (NOVO, 2010, p.79-80).

Já a resolução espectral se refere aos diferentes comprimentos das bandas espectrais que o sensor opera, estes podem ser medidos em micrômetros (μm) ou nanômetros (nm). Para se obter a identificação mais confiável de alvos específicos, a resolução espectral do sensor deve corresponder a mais possível curva de refletância espectral do alvo pretendido Mather (2004, p.32). Novo (2010, p.85) apresenta as vantagens de equipamentos com melhor resolução espectral, ao afirmar que: "A consequência do desenvolvimento de sensores com melhor resolução espectral é poder aumentar o número de bandas disponíveis para análise das iterações entre a radiação eletromagnética e os materiais que compõem a superfície terrestre".

Segundo Mather (2004, p.33), a resolução radiométrica, ou sensibilidade radiométrica, se refere: "ao número de níveis de quantização digital¹ usado para expressar os dados coletados pelo sensor. Em geral quanto maior o número de níveis de quantização, maior o detalhe da informação coletada pelo sensor".

Por fim, Mather (2004, p.25) faz referência à quarta característica, a resolução temporal, definindo essa característica como a periodicidade com que o sensor realiza a aquisição de informações em um determinado ponto do solo.

O *Sentinel-2* utiliza um sensor chamado MSI, ele consegue captar 13 diferentes bandas espectrais, sendo 4 delas em resolução espacial de 10m, 6 com resolução espacial de 20m e 3 com resolução espacial de 60m, conforme pode ser observado na Tabela 3. Além disso, o *Sentinel-2* apresenta resolução temporal de 10 dias e, com os dois satélites operacionais, essa resolução pode diminuir para 5 dias (ENGESAT, 2019b).

Já o *Landsat-8* possui outros sensores. São estes o PAN para resoluções espaciais de 15 metros e o MS para resoluções espaciais de 30 metros. Suas bandas espectrais são apresentadas

¹ A quantização pode ser definida como: "o mapeamento do sinal contínuo da cena em um número discreto de pontos organizados espacialmente (pixels), cada um com uma capacidade finita de representação (profundidade de pixel)". (SOLOMON; BRECON, 2013)

Tabela 3 – Tabela de bandas espectrais do *Sentinel-2*

Bandas	Comprimento de Onda Central (nm)	Resolução
B02 - Azul	490 nm	10m
B03 - Verde	560 nm	10m
B04 - Vermelho	665 nm	10m
B08 - Infravermelho próximo (NIR)	842 nm	10m
B05 - Borda Vermelha 1	705 nm	20m
B06 - Borda Vermelha 2	740 nm	20m
B07 - Borda Vermelha 3	783 nm	20m
B08A - Borda Vermelha 4	865 nm	20m
B11 - SWIR 1	1610 nm	20m
B12 - SWIR 2	2190 nm	20m
B01 - Aerossol	443 nm	60m
B09 - Vapor de Água	940 nm	60m
B10 - Cirrus	1375 nm	60m

Fonte: Adaptado de EngeSat (2019b)

na Tabela 4 (ENGESAT, 2019a).

Tabela 4 – Tabela de bandas espectrais do Landsat-8

Bandas	Comprimento de Onda	Resolução
Banda 1 - Aerossol Costeiro	430-450 nm	30 m
Banda 2 - Azul	450-510 nm	30 m
Banda 3 - Verde	530-590 nm	30 m
Banda 4 - Vermelho	640-690 nm	30 m
Banda 5 - Infravermelho próximo (NIR)	850-880 nm	30 m
Banda 6 - SWIR 1	1570-1650 nm	30 m
Banda 7 - SWIR 2	2110-2290 nm	30 m
Banda 8 - Pancromático	500-680 nm	15 m
Banda 9 - Cirrus	1360-1380 nm	30 m

Fonte: Adaptado de EngeSat (2019a)

3.2 MÉTODOS

3.2.1 PRÉ-PROCESSAMENTO

O primeiro passo necessário para esse trabalho foi a aquisição dos conjuntos de imagens do *BigEarthNet*. Após essa etapa foi necessário realizar a conversão desses dados conforme estabelecido na Tabela 2. Para a essa conversão foi utilizada a linguagem de programação *Python*, sendo necessária a criação de um *script* que no momento de sua execução consiga ler o arquivo no formato *JSON*, e redefinir as classes especificadas para cada imagem.

Em seguida, foi então criada uma rede neural convolucional capaz de realizar a classificação dessas imagens obtidas no *BigEarthNet*.

3.2.2 TREINAMENTO

Para o treinamento foi reservado um total de 70% dos conjuntos de imagens do *BigEarthNet*, isso é, um total de 354.196 conjuntos de *labels* e para a validação foi reservado um total de 30% que corresponde a 118.064 *labels*.

Será utilizada a linguagem de programação *Python* com a biblioteca *Keras*, que já fornece um conjunto de funções que podem ser reaproveitadas, como os métodos para criação de camadas convolucionais, camadas totalmente conectadas, adição de *max-pooling* entre as camadas e a adição de uma função *soft-max* na camada de saída.

Para o treinamento foi utilizada uma placa de vídeo *NVIDIA GEFORCE 940MX* de 4GB de memória de um computador *Acer Aspire F15*.

3.2.3 TESTES

A fase de testes foi realizada utilizando imagens do *Landsat-8* referentes ao estado de Goiás, que foram adquiridas na plataforma *EarthExplorer* e rotuladas de acordo com as classes do IBGE (2018).

A validação do modelo foi realizada utilizando as seguintes métricas: *recall* e *F1 Score*. Onde, foi utilizado o *recall* devido a quantidade diferente de imagens em cada uma das classes e o *F1 Score* devido às várias classes esperadas na saída para o modelo.

4 DESENVOLVIMENTO

4.1 IMAGENS DE TREINAMENTO E VALIDAÇÃO

O conjunto de imagens obtidas a partir da base de dados do *BigEarthNet*, é composto por doze diferentes arquivos *GeoTIFF* para as respectivas bandas espectrais encontradas no *Sentinel-2* e um arquivo com os metadados da imagem.

O diretório de arquivos está organizado de acordo com o nome da missão do *Sentinel-2* (S2A e S2B), concatenado com a sigla MSIL2A que representa o nível dos produtos 2A do *Sentinel-2*, a data da aquisição no formato ano/mês/dia, a hora da aquisição no formato hora/minuto/segundo, a coordenada no eixo horizontal e por fim a coordenada no eixo vertical de onde foi extraída a imagem, como pode ser visto na Figura 4.

Figura 4 – Organização dos diretórios do BigEarthNet

S2A_MSIL2A_20170613T101031_19_70	12/09/2019 16:10	Pasta de arquivos
S2A_MSIL2A_20170613T101031_19_73	12/09/2019 16:10	Pasta de arquivos
S2A_MSIL2A_20170613T101031_19_89	12/09/2019 16:10	Pasta de arquivos
S2A_MSIL2A_20170613T101031_20_35	12/09/2019 16:10	Pasta de arquivos

Fonte: O autor

Dentro do diretório estão organizadas as bandas espectrais separadas em arquivos no formato de *GeoTIFF*, georreferenciados como uma imagem no formato *raster*. Seus nomes seguem a definição de nomes do diretório com a concatenação do nome da banda espectral ao final do arquivo. Também se encontra um arquivo no formato JSON (*JavaScript Object Notation*) responsável por armazenar os metadados e os múltiplos rótulos vinculados a imagem, como pode ser visto na Figura 5.

Figura 5 – Arquivos disponibilizados pelo BigEarthNet

Nome	Tipo
S2A_MSIL2A_20170613T101031_19_70_B01.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B02.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B03.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B04.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B05.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B06.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B07.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B08.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B8A.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B09.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B11.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_B12.tif	Arquivo TIF
S2A_MSIL2A_20170613T101031_19_70_labels_metadata.json	Arquivo JSON

Fonte: O autor

Como entrada para o treinamento da rede convolucional foi necessária a criação de um conversor que recebe as imagens referentes às bandas vermelha, verde e azul e cria, a partir delas, um mosaico que é finalmente convertido para uma imagem *JPG* com as definições de cores no formato RGB (*Red, Green, Blue*).

O conversor foi desenvolvido utilizando a linguagem de programação *Python* e a biblioteca GDAL¹. Entre as linhas 1-7 foram importadas as bibliotecas necessárias para a correta execução do programas, conforme pode ser visto na Figura 6.

Figura 6 – Linhas de importação das bibliotecas

```

1  import os
2  import sys
3  import json
4  from osgeo import gdal
5
6  sys.path.append('C:\\Program Files (x86)\\GDAL')
7  import gdal_merge as gm

```

Fonte: O autor

A linha 9 define o diretório que será percorrido, enquanto que nas linhas de 11-16 são realizadas as configurações para conversão do *TIF* para o *JPG*. Após isso, na linha 19 o algoritmo percorre todas as pastas que estão presentes no diretório raiz, na linha 19 são percorrido todos os arquivos que estão presentes dentro das pastas e, por fim, na linha 20 é limitado o número de vezes que a conversão é realizada por pasta. As linhas 21-25 são usadas para definir o caminho até a pasta para gerar a imagem no lugar correto. E as linhas 28-38 são responsáveis pela criação do mosaico, conforme pode ser visto na Figura 7.

Figura 7 – Criação do mosaico e do arquivo TIF

```

28  gm.main([
29      '',
30      '-init',
31      '255 255 255',
32      '-separate',
33      '-o',
34      dir_nova_imagem + '.tif',
35      dir_imgs_rgb + '_B04.tif',
36      dir_imgs_rgb + '_B03.tif',
37      dir_imgs_rgb + '_B02.tif'
38  ])

```

Fonte: O autor

Nas linhas 42-45 são definidas as cores de cada camada do mosaico. As linhas 48-52 são responsáveis por converter o mosaico no formato *TIF* para uma imagem *JPG*. E por fim, as linhas 55-57 removem os arquivos que não são necessários da pasta, conforme apresentado na Figura 8.

¹ GDAL: <<https://gdal.org/index.html>>

Figura 8 – Conversão do TIF para JPG

```

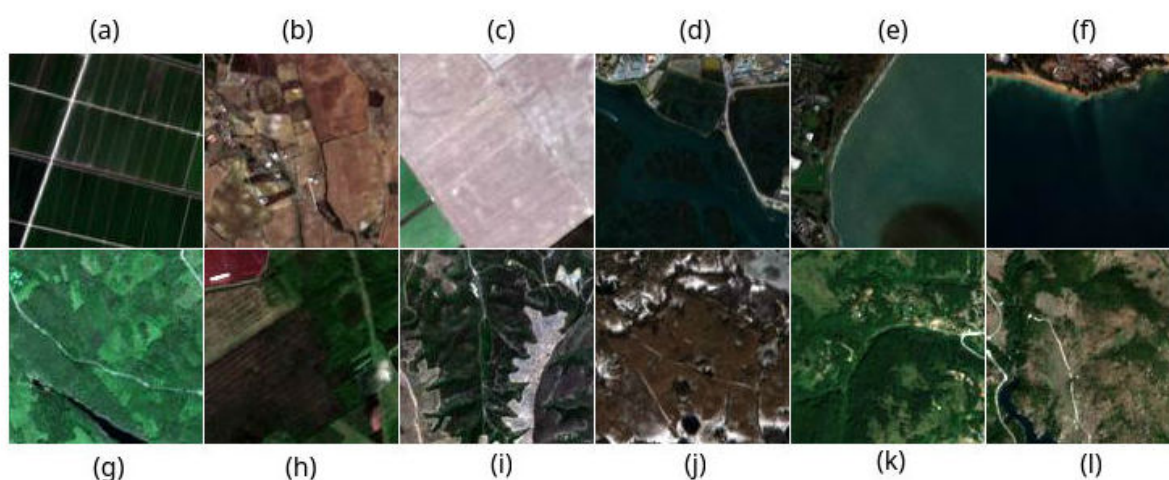
40 # Muda a definição das bandas da figura.
41 raster = gdal.Open(dir_nova_imagem + '.tif')
42 raster.GetRasterBand(1).SetColorInterpretation(gdal.GCI_RedBand)
43 raster.GetRasterBand(2).SetColorInterpretation(gdal.GCI_GreenBand)
44 raster.GetRasterBand(3).SetColorInterpretation(gdal.GCI_BlueBand)
45 del raster
46
47 # Transforma a figura ".tif" em uma figura ".jpg"
48 gdal.Translate(
49     dir_nova_imagem + '.jpg',
50     dir_nova_imagem + '.tif',
51     options=options_string
52 )
53
54 # Remove os arquivos desnecessários
55 os.remove(dir_nova_imagem + '.tif')
56 os.remove(dir_nova_imagem + '.tif.aux.xml')
57 os.remove(dir_nova_imagem + '.jpg.aux.xml')

```

Fonte: O autor

A Figura 9 apresenta alguns exemplos de imagens que foram geradas a partir dessa conversão. A classe de área agrícola é representada nas três primeiras figuras, porém, é perceptível a diferença entre elas, a Figura 9 (a) apresenta uma área verde com um padrão de plantio bem definido, já a Figura 9 (b) apresenta uma área para plantio sem plantação e com o solo em tonalidade vermelha, por fim a Figura 9 (c) apresenta uma área sem nenhum cultivo, porém com uma coloração mais clara para o solo.

Figura 9 – Exemplo de imagens geradas na conversão



Fonte: O autor

A Figura 9 (d), Figura 9 (e) e Figura 9 (f) representa exemplos de áreas artificiais mescladas com corpos d'água costeiros, novamente é possível perceber a variedade de elementos que podem ser encontrados em uma mesma classe. Onde a coloração do mar pode ser diferente, os elementos da área artificial podem estar mesclados com outros elementos naturais, e o ponto de encontro entre o mar e o solo apresenta diferenças de coloração bastante visíveis.

A Figura 9 (g), Figura 9 (h) e Figura 9 (i) apresentam exemplos da mescla entre vegetação campestre e vegetação florestal. Na Figura 9 (g) existe uma dificuldade em perceber a diferença entre essas classes. Já na Figura 9 (h), essa diferença é perceptível por causa da tonalidade dos elementos. E na Figura 9 (i) tem-se algo totalmente diferente das duas anteriores, onde a vegetação florestal e campestres estão distribuídas ao longo de uma cadeia montanhosa.

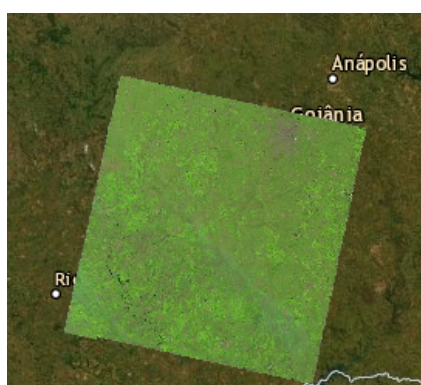
Por fim, as Figuras 9 (j), (k), (l) apresentam a grande possibilidade de mistura entre as classes, visto que elas representam: áreas artificiais, corpo d'água continental, mosaicos de ocupação em área florestal, pastagem com manejo, vegetação campestre e vegetação florestal. É perceptível nessas comparações que o uso e cobertura do solo em classes iguais se comporta de diferentes formas e apresentam diversas variações entre os *pixels* e elementos perceptivos das figuras.

4.2 IMAGENS DE TESTE

O conjunto de imagem para teste da rede neural artificial, foi obtido manualmente com o auxílio de ferramentas como o *ArcMap 10.5*, o site do *Earth Explorer* pertencente ao *U.S. Geological Survey (USGS)* e o mapa iterativo de cobertura e uso do solo disponível pelo Sistema Estadual de Geoinformação do estado de Goiás (SIEG). Foi utilizada uma imagem referente a Microrregião Meia Ponte, essa microrregião se encontra ao sul do estado e todas as imagens foi obtidas a partir do satélite *Landsat-8*.

O *download* da imagem foi realizado através do site do *Earth Explorer*. Na aba *Data Sets* foi selecionado o *Landsat 8 OLI/TIRS C1 Level-1*. Como critérios adicionais foram adicionadas as coordenadas de órbita e ponto do Sistema de Referência Mundial (*The Worldwide Reference System, WRS*) para seleção apenas das coordenadas referentes ao estado de Goiás. Outro critério adotado foi verificar a presença de nuvens, com quantidade inferior a 90% para cobertura de terra por nuvens e cobertura da cena por nuvens. O *GeoTiff* obtido pode ser visto na Figura 10.

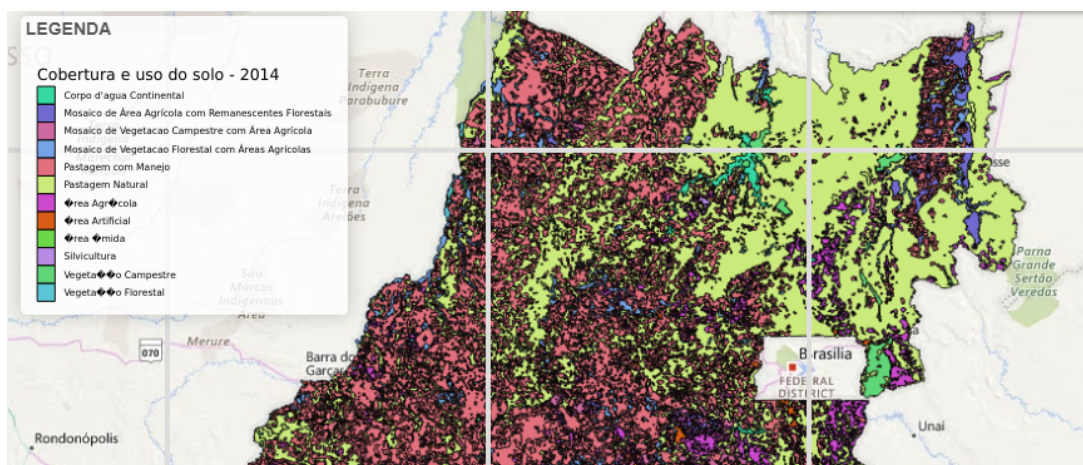
Figura 10 – *Geotiff* utilizado para montar o conjunto de teste



Fonte: Adaptado de USGS (2019)

O site do SIEG foi utilizado para realizar o *download* do *shapefile*² com a classificação de cobertura e uso do solo do Estado de Goiás, segundo as diretrizes do IBGE. Esse *shapefile* é um arquivo importante, pois ele apresenta a classificação já feita por profissionais qualificados. Como pode ser visto na Figura 11.

Figura 11 – Classificação da Cobertura e Uso do Solo do estado de Goiás



Fonte: (SIEG, 2019)

Com esses dois arquivos e com o auxílio do *ArcMap 10.5* teve início a criação de polígonos que foram utilizados para recortar o arquivo *Geotiff* e gerar a massa de dados para teste. Foram criados 150 polígonos para esse fim, que foram posicionados sobre o *Geotiff* de maneira a abranger uma área de aproximadamente $130 * 130$ pixels.

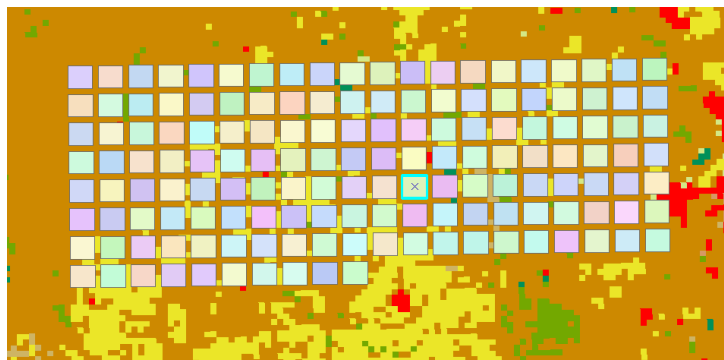
Foi realizado, então, o processo de *clip* do *Geotiff* utilizando como arquivo de corte os polígonos gerados. Esse processo gerou novos *Geotiffs* específicos para a área do polígono. Após esse processo de *clip*, foi utilizada uma ferramenta de exportação de dados presente no *ArcMap* para geração dos arquivos físicos *TIF* referentes as áreas recortadas. Esses arquivos *TIF* foram convertidos para o formato *JPG* para ficarem de acordo com as imagens de entrada para o treinamento e validação da rede. Devido a ruídos encontrados em algumas imagens, foram utilizadas 138 do total de 150 para o teste da rede neural.

Sobrepondo o *shapefile* de classificação do uso e cobertura do solo do Estado de Goiás com os polígonos gerados foi possível determinar qual a classificação de cada polígono de acordo com a cor atribuída para a classificação do *shapefile*, como pode ser visto na Figura 12.

Após a etapa de classificação de cada uma das imagens geradas pelos polígonos, foi necessário o agrupamento e a criação de diretórios específicos para cada imagem. Esses diretórios foram criados a partir dos nomes das classes, em ordem alfabética, sendo esses nomes concatenados por um traço que indica a separação entre elas.

² Formato popular de arquivo contendo dados geoespaciais em forma de vetor usado por Sistemas de Informações Geográficas também conhecidos como SIG.

Figura 12 – Sobreposição dos polígonos no *shapefile*



Fonte: O autor

A região utilizada como massa para teste possui o predomínio das classes de pastagem com manejo e área agrícola, possuindo também uma diversidade em mosaico de ocupação em área florestal, mosaico de ocupação em área campestre, vegetação florestal, vegetação campestre, área artificial e silvicultura. Alguns exemplos dessas imagens para teste podem ser vistos na Figura 13.

Figura 13 – Exemplo de imagens do conjunto de teste



Fonte: O autor

4.3 TREINAMENTO DA REDE NEURAL ARTIFICIAL

Para o treinamento da rede foi realizada a conversão entre as classes encontradas no conjunto de dados do *BigEarthNet* para suas respectivas classes definidas pelo IBGE. Foi utilizado para essa conversão um novo *script* desenvolvido na linguagem de programação *python*, responsável por ler o arquivo no formato *JSON*, interpretar os metadados e reescrever nos metadados na sequência de classes respectivas do IBGE.

Essa conversão seguiu as normalizações definidas na Tabela 2. O *script* percorreu cada definição do objeto *labels* no metadado, e verificando um a um foi criando uma nova definição de objeto, chamada de *labels_ibge*.

Para servir de dado de entrada para a rede, foi realizada a separação de cada conjunto de classes em um diretório específico. Dessa forma, todas as imagens vinculadas a uma classe foram para o diretório referente aquela classe. Houveram cenários especiais onde uma imagem possui mais de uma classe, para esses cenários foram criados diretórios contendo o nome de todas as classes vinculadas, separadas por um traço. Dessa forma, foram obtidas 12 classes gerais

e 496 combinações de classes. Esse conjunto com 508 combinações entre as classes serviu como entrada na fase de treinamento da rede neural artificial.

Foi criado um modelo para realização dos testes com o conjunto de dados de imagens adquirido. Esse modelo tem como primeira camada uma convolução, como entrada, essa camada convolucional recebe uma matriz de valores inteiros que representam os *pixels* da imagem. Cada imagem fornece uma matriz de $120 * 120 * 3$, sendo os dois primeiros valores referentes a altura e largura da imagem e o terceiro valor referente a quantidade de bandas que essa imagem possui. A função de ativação utilizada nessa camada foi a *relu*. A próxima camada da rede é uma camada de agrupamento. A terceira camada é outra camada convolucional que recebe como entrada a saída da camada de agrupamento, a função de ativação utilizada nessa camada foi a *relu*. A próxima camada da rede é uma camada de agrupamento. A quinta camada é outra camada de convolução, que recebe como entrada a saída da camada de agrupamento anterior, a função de ativação utilizada é a *relu*. A sexta camada é outra camada de agrupamento. Por fim, a sétima camada é uma camada totalmente agrupada, e a oitava camada é a camada de saída que define as 12 classes do modelo que está sendo treinado e tem como função de ativação a *sigmoid*, isso é, uma função de ativação utilizada para determinar saídas multi-classes.

A rede neural artificial recebeu um total de 70.323 imagens para o treinamento e validação. Sendo 70% desse total reservado para o treinamento e 30% reservado para o conjunto de validação. Para a primeira execução da rede neural utilizou-se da estratégia de carregar todas essas imagens de uma vez na memória do computador, isso ocasionou uma sobrecarga na RAM e no *SWAP* de memória. Foi então utilizada a estratégia de carregar as imagens sob demanda utilizando o carregamento em lotes disponibilizados pela classe *ImageDataGenerator* do *Keras* e pelo método *flow_from_directory* pertencente essa classe.

Outra abordagem utilizada foi a criação de *checkpoints* que foram informados para o métodos de treinamento da rede, eles são responsáveis por salvar os melhores pesos encontrados na rede para a acurácia e para a função de perda. Esses *checkpoints* foram repassados para a função de retorno do treinamento.

4.3.1 RESULTADOS DO TREINAMENTO E VALIDAÇÃO

O treinamento ocorreu em 15 ciclos, conhecidos como épocas, para cada uma dessas épocas os pesos da rede eram reajustados de acordo com as imagens de entrada. Além disso, após cada ciclo foi gerado uma porcentagem referente aos valores de acurácia, precisão, *recall* e *F1 score*. A Tabela 5 apresenta as porcentagens de cada uma dessas medidas para cada uma das épocas na etapa de treinamento. Já os resultados encontrados durante a validação da rede, podem ser observados na Tabela 6.

Os valores considerados nesse modelo, foram os referentes ao *recall* e ao *F1 score*. Devido ao desbalanceamento existente entre as classes e também às várias classes esperadas na saída.

Tabela 5 – Resultado para cada época do treinamento

Época	Acurácia	Precisão	Recall	F1 Score
1	27,07%	65,66%	50,25%	55,83%
2	37,59%	77,42%	44,34%	55,95%
3	39,45%	81,02%	43,79%	56,41%
4	40,97%	82,47%	45,73%	58,38%
5	41,76%	82,52%	50,31%	62,06%
6	42,10%	80,97%	55,83%	65,76%
7	42,54%	80,13%	61,08%	69,01%
8	42,69%	79,36%	65,42%	71,48%
9	42,86%	78,75%	69,04%	73,35%
10	43,11%	78,17%	72,39%	74,97%
11	42,96%	77,84%	74,95%	76,18%
12	42,84%	77,54%	77,31%	77,25%
13	43,19%	77,68%	79,25%	78,29%
14	43,11%	77,52%	81,02%	79,08%
15	43,07%	77,49%	82,58%	79,81%

Fonte: O autor

Tabela 6 – Resultado para cada época da validação

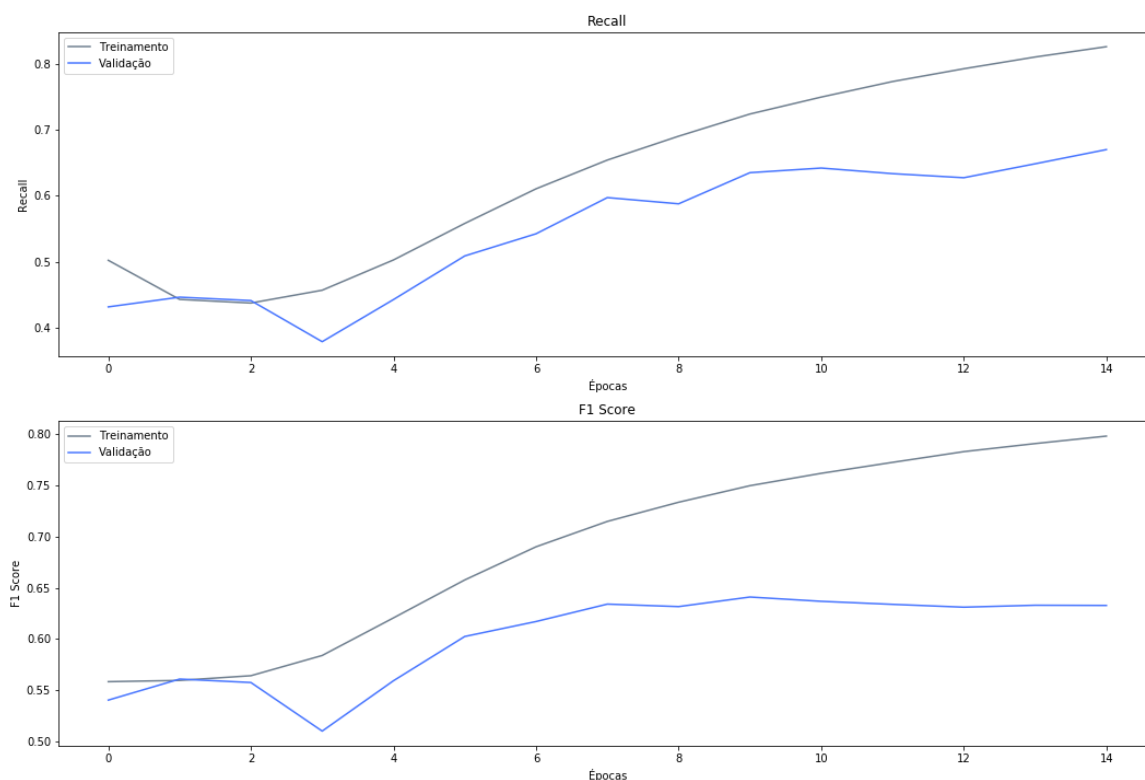
Época	Acurácia	Precisão	Recall	F1 Score
1	47,82%	72,81%	43,20%	54,03%
2	43,82%	76,04%	44,68%	56,08%
3	33,20%	76,49%	44,16%	55,75%
4	43,29%	79,03%	37,94%	51,00%
5	37,67%	76,69%	44,31%	55,92%
6	42,99%	74,41%	50,92%	60,24%
7	45,28%	72,09%	54,27%	61,70%
8	45,71%	67,95%	59,75%	63,40%
9	41,44%	68,59%	58,80%	63,16%
10	40,30%	64,98%	63,52%	64,10%
11	38,19%	63,42%	64,22%	63,68%
12	34,01%	63,66%	63,37%	63,38%
13	37,59%	63,76%	62,75%	63,10%
14	38,01%	62,04%	64,86%	63,29%
15	37,20%	60,16%	67,01%	63,27%

Fonte: O autor

Os pesos da rede neural salvos foram os que atingiram o maior *F1 score* na etapa de validação. Pode-se então concluir que a rede atingiu uma porcentagem de acerto no *recall* de 63,52% e um *F1 score* de 64,10%.

Outro ponto importante para se verificar é a diferença de resultados entre o *recall* e o *F1 score* com relação às etapa de treinamento e de validação, sendo os resultados apresentados na etapa de validação inferiores aos obtidos no treinamento. Conforme pode ser visualizado melhor nos gráficos apresentados na Figura 14. Esse resultado pode ter sido gerado por causa do conjunto de imagens que foi aleatoriamente escolhido para a validação. Onde, uma determinada classe teve maior representação no treinamento e pouca representação na validação, ou o efeito contrário, onde uma classe teve muita ocorrência na validação mas pouca ocorrência no treinamento.

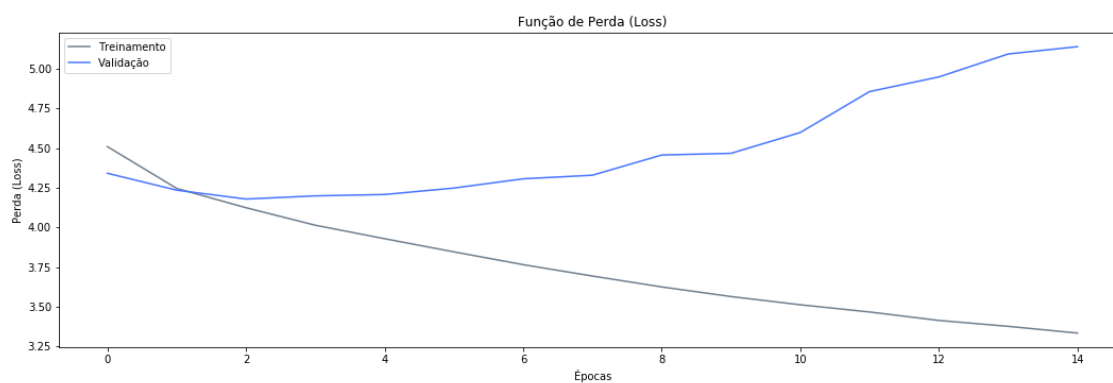
Figura 14 – Resultados do treinamento e da validação para o *recall* e *F1 score*



Fonte: O autor

Outro parâmetro que também precisa de atenção é a função de perda. Ela é responsável por reduzir o erro na predição dos resultados e dessa forma apresentar maior acerto nos resultados. Porém, a função de perda que deveria reduzir seu valor a cada época, teve esse valor aumentado para o conjunto de validação, conforme pode ser visto na Figura 15. Dessa forma, é possível deduzir que este foi um dos motivos pelos quais o resultado do conjunto de validação foi inferior ao do conjunto de treinamento.

Figura 15 – Função de perda no treinamento e validação



Fonte: O autor

4.3.2 RESULTADOS DO TESTE

O teste realizado nas imagens obtidas no sul do Estado de Goiás apresentou resultados interessantes. Obteve-se um *recall* de 28,67% e um F1 score de 21,47%. Vários fatores podem ter levado a esse resultado.

Um desses fatores é o *overfitting* no treinamento da rede. Existem muitas combinações entre as classes e poucas representações para algumas dessas combinações, o que ocasiona um problema de desbalanceamento entre o números de imagens utilizado para o treinamento ou validação da rede. Dessa forma, algumas classes mais gerais possuem até 1000 imagens que as representam, enquanto que algumas classes mais específicas possuem apenas uma imagem para essa representação.

Outro fator que pode ter ocasionado esse resultado nos dados de teste é a semelhança de padrões entre algumas classes apresentadas. Como pode ser visto em área agrícola, mosaico de ocupação em área florestal e pastagem com manejo, por exemplo. A Figura 16 apresenta essa semelhança. As duas primeiras imagens representam área agrícola, as imagens três e quatro representam mosaico de ocupação em área florestal e as duas últimas representam pastagem com manejo.

Figura 16 – Semelhança entre as classes



Fonte: O autor

Por fim, outro fator que pode ter ocasionado esse resultado está relacionado aos dados de entrada para a rede neural serem imagens no formato RGB. Ao decidir eliminar a utilização de outras bandas das imagens de satélites, e dados que poderiam ter sido utilizados para apresentar outros padrões que a rede não conseguiu aprender foram perdidos, ficando essa limitada apenas à três bandas.

5 CONSIDERAÇÕES FINAIS

No desenvolvimento desse trabalho foram encontradas algumas dificuldades ao se manusear dados de satélite em conjunto com redes neurais convolucionais para a classificação de cobertura e uso do solo.

O processo de obtenção de imagens de satélites foi facilitado devido à utilização do conjunto de dados do *BigEarthNet*. Esse conjunto de dados possui uma grande variedade de classes de cobertura e uso do solo. Um problema encontrado no *BigEarthNet* foi a falta de balanceamento entre a quantidade de elementos por classe. Já a obtenção do conjunto de teste foi bastante onerosa em questão de tempo, devido a necessidade de realizar muito trabalho manual para o recorte das imagens com os polígonos e para a classificação do tipo de cobertura e uso de solo que ela representa.

Foi desenvolvida uma rede neural convolucional com poucas camadas para realizar o teste de classificação desse conjunto de dados. O desenvolvimento de uma rede mais complexa, junto com um melhor balanceamento dos dados entre as classes poderia apresentar resultados mais interessantes para esse tipo de problema.

O treinamento da rede neural convolucional apresentou resultados bastante significativos, porém, esses resultados podem ter sido gerados por causa de *overfitting* na rede, isso é, a rede se especializou nos dados do conjunto de treinamento e não conseguiu generalizar o suficiente para outros tipos de dados.

A forma como o conjunto de treinamento e validação foi informado para a rede neural convolucional talvez tenha sido um dos fatores que gerou o resultado obtido na etapa de testes com imagens do Estado de Goiás.

Alguns outros problemas podem ter gerado essa baixa porcentagem de acerto para o conjunto de teste. Entre esses problemas podem ser destacadas a semelhança de padrões entre algumas classes, as entradas serem imagens no formato RGB ou ainda o grande desbalanceamento entre as classes.

REFERÊNCIAS

- AMATO, F. et al. Remotely Sensed Soil Data Analysis Using Artificial Neural Networks: A Case Study of El-Fayoum Depression, Egypt. *ISPRS Int. J. Geo-Inf*, v. 4, p. 677–696, 2015. Citado na página 13.
- CHOLLET, F. *Deep Learning with Python*. Shelter Island, NY: Manning Publications Co., 2018. ISSN 9781617294433. Citado na página 27.
- CLC. *CORINE Land Cover*. [S.l.], 1995. Disponível em: <<https://www.eea.europa.eu/publications/COR0-landcover>>. Acesso em: 18 de maio de 2019. Citado na página 13.
- CLC. *Implementation and achievements of CLC2006*. [S.l.], 2006. Disponível em: <https://land.copernicus.eu/user-corner/technical-library/CLCFinalrep_revised_finaldraft.pdf>. Acesso em: 16 de junho de 2019. Citado na página 30.
- COPPIN, B. *Inteligência Artificial*. 1. ed. Rio de Janeiro: LTC, 2013. 636 p. ISSN 978-85-216-1729-7. Citado na página 16.
- CORRÊA, I. et al. Aprendizado Profundo Aplicado na Classificação de Microalgas. *CEIA-SBC*, Sociedade Brasileira de Computação, v. 14, p. 810–818, 2017. Citado na página 13.
- DATA SCIENCE ACADEMY. *Deep Learning Book: Capítulo 3 – O Que São Redes Neurais Artificiais Profundas ou Deep Learning?* 2017. Disponível em: <<http://deeplearningbook.com.br/o-que-sao-redes-neurais-artificiais-profundas>>. Acesso em: 18 de maio de 2019. Citado na página 23.
- ELAINE, R.; KNIGHT, K.; NAIR, S. B. *Artificial Intelligence*. 3. ed. New Delhi: The McGraw-Hill Companies, 2009. 421 p. ISSN 978-0-07-008770-5. Citado na página 16.
- ENGESAT. *LANDSAT 8*. 2019. Disponível em: <<http://www.engesat.com.br/imagem-de-satelite/landsat-8/>>. Acesso em: 30 de abril de 2019. Citado na página 32.
- ENGESAT. *SENTINEL-2*. 2019. Disponível em: <<http://www.engesat.com.br/sentinel-2>>. Acesso em: 16 de junho de 2019. Citado 2 vezes nas páginas 31 e 32.
- FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro: LTC, 2011. 421 p. ISSN 978-85-216-188000-5. Citado 5 vezes nas páginas 19, 20, 21, 22 e 23.
- FAEG. *Relatório Anual 2017*. [S.l.], 2017. Disponível em: <<http://sistemafaeg.com.br/public/uploads/db3f96ae336bbd70e53ccd5d982953b5.pdf>>. Acesso em: 18 de junho de 2019. Citado na página 29.
- FILHO, O. M.; NETO, H. V. *Prossamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. 331 p. ISSN 8574520098. Citado na página 15.
- FRED, L. Machine Learning: a matemática da aprendizagem supervisionada. Medium, 11 2017. Disponível em: <<https://medium.com/luisfredgs/machine-learning-a-matemática-da-aprendizagem-supervisionada-6465f46b3a>>. Acesso em: 18 de maio de 2019. Citado na página 22.

- GONZALEZ, R. C.; WOODS, R. C. *Processamento Digital de Imagens*. 3. ed. São Paulo: Pearson, 2010. 644 p. Citado na página 15.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: <<http://www.deeplearningbook.org>>. Citado 3 vezes nas páginas 20, 24 e 25.
- HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and techniques*. 3. ed. Waltham, MA: Elsevier Inc., 2018. ISSN 978-0-12-381479-1. Citado 2 vezes nas páginas 26 e 27.
- HAYKIN, S. *Redes Neurais: Princípios e práticas*. 2. ed. Porto Alegre: Bookman, 2001. 900 p. ISSN 978-85-7307-718-6. Citado 3 vezes nas páginas 21, 22 e 23.
- HUI, Y.; WIELE, C. V. D.; SIAMAK, K. An Automated Artificial Neural Network System for Land Use/Land Cover Classification from Landsat TM Imagery. *Remote Sensing*, v. 1, setembro de 2009. Citado na página 13.
- IBGE. *MONITORAMENTO DA COBERTURA E USO DA TERRA DO BRASIL 2014 - 2016*. Rio de Janeiro, 2018. Disponível em: <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv101625.pdf>>. Acesso em: 15 de abril de 2019. Citado 5 vezes nas páginas 13, 29, 30, 33 e 50.
- IBGE. *Monitoramento da Cobertura e Uso da Terra | IBGE*. 2019. Disponível em: <<https://www.ibge.gov.br/geociencias/informacoes-ambientais/cobertura-e-uso-da-terra/15831-cobertura-e-uso-da-terra-do-brasil.html?=&t=o-que-e>>. Acesso em: 18 de maio de 2019. Citado na página 13.
- JUNIOR, T. F. T. Dimas Tadeu de O. Classificação de Gêneros de Música Brasileira usando Letras e Acordes. *CEIA-SBC*, Sociedade Brasileira de Computação, v. 14, p. 496–506, 2017. Citado na página 13.
- KELLEHER, J. D.; NAMEE, B. M.; D'ARCY, A. *Fundamentals of Machine Learning For Predictive Data Analytics: Algorithms, worked examples, and case studies*. London, England: The MIT Press, 2015. ISSN 978-0-262-02944-5. Citado na página 26.
- MATHER, P. M. *Computer Processing of Remotely-Sensed Images: An Introduction*. 3. ed. Chichester: John Wiley & Sons Ltd, 2004. ISSN 0-470-84919-3. Citado na página 31.
- MENESES, P. R.; ALMEIDA, T. d. *INTRODUÇÃO AO PROCESSAMENTO DE IMAGENS DE SENSORIAMENTO REMOTO*. Brasília: UnB e CNPq, 2012. Disponível em: <<http://cnpq.br/documents/10157/56b578c4-0fd5-4b9f-b82a-e9693e4f69d8>>. Citado na página 16.
- MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 19.
- MONTABONE, S. *Beginning Digital Image Processing: Using Free Tools for Photographers*. [S.l.]: Apress, 2010. 312 p. ISSN 978-1-4302-2841-7. Citado na página 15.
- NOVO, E. M. L. D. M. *Sensoriamento Remoto Princípios e Aplicações*. 4. ed. São Paulo: Blucher, 2010. ISSN 978-85-212-0540-1. Citado na página 31.
- POWERS, D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Makedness & Correlation. *Journal of Machine Learning Technologies*, 2015. ISSN 2229-3981. Acesso em: 14 de junho de 2019. Citado na página 27.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. 3. ed. Rio de Janeiro: Elsevier, 2013. 988 p. ISSN 978-85-352-3701-6. Citado 4 vezes nas páginas 17, 18, 19 e 22.

SIEG. *SIEG Downloads*. 2019. Disponível em: <<http://www.sieg.go.gov.br/siegdownloads/>>. Acesso em: 03 de novembro de 2019. Citado na página 38.

SOLOMON, C.; BRECON, T. *Fundamentos de Processamento Digital de Imagens - Uma Abordagem Prática com Exercícios em Matlab*. 1. ed. Rio de Janeiro: LTC, 2013. 288 p. Citado 4 vezes nas páginas 15, 16, 17 e 31.

SUMBUL, G. et al. BigEarthNet: A large-scale benchmark archive for remote sensing image understanding. arXiv preprint, 2019. Disponível em: <<https://arxiv.org/pdf/1902.06148.pdf>>. Acesso em: 20 de maio de 2019. Citado 2 vezes nas páginas 29 e 30.

SUMMERFIELD, M. *Programming in Python 3: A complete introduction to the python language*. 2. ed. Upper Saddle River, NJ: Addison Wesley, 2018. ISSN 978-0-321-68056-3. Citado na página 28.

TURING, A. M. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX, n. 236, p. 433–460, 10 1950. ISSN 0026-4423. Disponível em: <<https://doi.org/10.1093/mind/LIX.236.433>>. Citado na página 18.

USGS. *USGS.gov | Science for a changing world*. 2019. Disponível em: <<https://www.usgs.gov>>. Acesso em: 18 de maio de 2019. Citado 2 vezes nas páginas 13 e 37.

VASILEV, I. et al. *Python Deep Learning: Exploring deep learning techniques and neural network architectures with pytorch, keras, and tensorflow*. 2. ed. Birmingham, Mumbai: Packt, 2019. ISSN 978-1-78934-846-0. Citado 5 vezes nas páginas 23, 24, 25, 26 e 28.

WITTER, I.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. 2. ed. San Francisco, CA: Elsevier, 2005. ISSN 0-12-088407-0. Citado na página 20.

Anexos

ANEXO A – CLASSES DE COBERTURA E USO DA TERRA - IBGE

CLASSES DE COBERTURA E USO DA TERRA

Cód.	Classe	Descrição
1	Área Artificial	Áreas onde predominam superfícies antrópicas não-agrícolas. São aquelas estruturadas por edificações e sistema viário, nas quais estão incluídas as metrópoles, cidades, vilas, as aldeias indígenas e comunidades quilombolas, áreas de rodovias, serviços e transportes, redes de energia, comunicações e terrenos associados, áreas ocupadas por complexos industriais e comerciais e edificações que podem, em alguns casos, estar situadas em áreas peri-urbanas. Também pertencem a essa classe as áreas onde ocorrem a exploração ou extração de substâncias minerais, por meio de lavra ou garimpo
2	Área Agrícola	Área caracterizada por lavouras temporárias, semi-perenes e permanentes, irrigadas ou não, sendo a terra utilizada para a produção de alimentos, fibras e commodities do agronegócio. Inclui todas as áreas cultivadas, inclusive as que estão em pousio ou localizadas em terrenos alagáveis. Pode ser representada por zonas agrícolas heterogêneas ou extensas áreas de plantations. Inclui também os tanques de aquicultura.
3	Pastagem com Manejo	Áreas destinadas ao pastoreio do gado e outros animais, com vegetação herbácea cultivada (braquiária, azevém, etc) ou vegetação campestre (natural), ambas apresentando interferências antrópicas de alta intensidade. Estas interferências podem incluir o plantio; a limpeza da terra (destocamento e despedramento); eliminação de ervas daninhas de forma mecânica ou química (aplicação de herbicidas); gradagem; calagem; adubação; entre outras que descaracterizem a cobertura natural.
4	Mosaico de Ocupações em Área Florestal	Área caracterizada por ocupação mista de agricultura, pastagem e/ou silvicultura associada ou não a remanescentes florestais, na qual não é possível uma individualização de seus componentes. Inclui também áreas com perturbações naturais e antrópicas, mecânicas ou não mecânicas, que dificultem a caracterização da área.
5	Silvicultura	Área caracterizada por plantios florestais de espécies exóticas ou nativas como monoculturas.
6	Vegetação Florestal	Área ocupada por florestas. Consideram-se florestais as formações arbóreas com porte superior a 5 metros de altura, incluindo-se aí as áreas de Floresta Ombrófila Densa, de Floresta Ombrófila Aberta, de Floresta Estacional, além da Floresta Ombrófila Mista. Inclui outras feições em razão de seu porte superior a 5 m de altura, como a Savana Florestada, Campinarana Florestada, Savana-Estépica Florestada, os Manguezais e os Buritizais, conforme o Manual Técnico de Uso da Terra (IBGE, 2013).

9	Área Úmida	Área caracterizada por vegetação natural herbácea (cobertura de 10% ou mais), permanentemente ou periodicamente inundada por água doce ou salobra. Inclui os terrenos de charcos, pântanos, campos úmidos, estuários, entre outros. O período de inundação deve ser de no mínimo 2 meses por ano. Pode ocorrer vegetação arbustiva ou arbórea, desde que estas ocupem área inferior a 10% do total.
10	Vegetação Campestre	Área caracterizada por formações campestres. Entende-se como campestres as diferentes categorias de vegetação fisionomicamente bem diversas da florestal, ou seja, aquelas que se caracterizam por um estrato predominantemente arbustivo, esparsamente distribuído sobre um estrato gramíneo-lenhoso. Incluem-se nessa categoria as Savanas, Estepes, Savanas Estépicas, Formações Pioneiras e Refúgios Ecológicos. Encontram-se disseminadas por diferentes regiões fitogeográficas, compreendendo diferentes tipologias primárias: estepes planáltinas, campos rupestres das serras costeiras e campos hidroarenosos litorâneos (restinga), conforme o Manual Técnico de Uso da Terra (IBGE, 2013). Essas áreas podem estar sujeitas a pastoreio e a outras interferências antrópicas de baixa intensidade como as áreas de pastagens não manejadas do Rio Grande do Sul e do Pantanal.
11	Mosaico de Ocupação em Área Campestre	Área caracterizada por ocupação mista de agricultura, pastagem e/ou silvicultura associada ou não a remanescentes campestres, na qual não é possível uma individualização de seus componentes. Inclui também áreas com perturbações naturais e antrópicas, mecânicas ou não mecânicas, que dificultem a caracterização da área.
12	Corpo d'água Continental	Inclui todas as águas interiores, como rios, riachos, canais e outros corpos d'água lineares. Também engloba corpos d'água naturalmente fechados (lagos naturais) e reservatórios artificiais (represamentos artificiais de água construídos para irrigação, controle de enchentes, fornecimento de água e geração de energia elétrica).
13	Corpo d'água Costeiro	Inclui todas as águas costeiras (lagunas, estuários e baías que ocupam as Planícies Costeiras) e as águas inseridas nas 12 milhas náuticas, conforme Lei nº 8.617, de 4 de janeiro de 1993.
14	Área Descoberta	Esta categoria engloba locais sem vegetação, como os afloramentos rochosos, penhascos, recifes e terrenos com processos de erosão ativos. Também inclui as praias e dunas, litorâneas e interiores, e acúmulo de cascalho ao longo dos rios.

Fonte: IBGE (2018)