DOI: 10.48011/asba.v2i1.1412

Análise de Parâmetros de Redes Neurais com Foco em Estabilidade: Estudo de Caso de Classificação de Defeitos em Chapas de Aço Através de Imagens

Alexandre R. Mundim * **Flávia M. F. Ferreira** *

* *Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-Graduação em Engenharia Elétrica, (e-mails: alexandremundim92@terra.com.br, flaviamagfreitas@gmail.com).*

Abstract: Stainless steel is one of the most important materials in the industry. Its features allow it to be used in several applications, being part in a series of products used in our daily lives. Stainless steel sheets are presented as one of the primary configurations of the material, standing out by its versatility. In this fashion, improvement on its manufacturing are extremely relevant for the guaranteeing of fast and quality supply. In this context, Artificial Intelligence algorithms - as neural networks - can significantly contribute to the betterment of the manufacturing processes, due to the elevated efficiency and assertiveness in the tasks they are submitted. Based on this assumption, Convolutional Neural Networks were applied to diagnose defects in stainless steel sheets. Even if these algorithms can present elevated performance in this task, its best performance is obtained through the exploration of an extensive parameter space, and application of regularization techniques. In scenarios with lesser data, these challenges are magnified. Thus, this paper present considerations concerning the influence of different adjustments in neural networks configuration with a focus on stability in the task of defect diagnosis on stainless steel sheets images.

Resumo: O aço inox é um dos materiais mais importantes da indústria. Suas propriedades permitem que seja utilizado em diversas aplicações, estando presente em uma série de produtos utilizados em nosso dia-a-dia. Chapas de aço inox se apresentam como uma das configurações básicas do material, se destacando pela sua versatilidade. Assim, melhorias em sua fabricação são de extrema relevância para garantia de seu fornecimento com agilidade e qualidade. Nesse contexto, algoritmos de inteligência artificial - a exemplo de redes neurais - podem contribuir significativamente para o aperfeiçoamento de seus processos de produção, por serem capazes de fornecer elevada eficiência e assertividade nas tarefas às quais são submetidos. Partindo desse pressuposto, utilizamos redes neurais convolucionais para diagnóstico de defeitos em chapas de aço através de imagens. Ainda que esses algoritmos possam apresentar elevada performance nesse tipo de tarefa, seu melhor desempenho é obtido mediante a exploração de um extenso espaço de parâmetros e aplicação de técnicas de regularização. Em cenários com dados reduzidos, esses desafios são ampliados. Dessa forma, o artigo apresenta considerações a respeito da influência de diferentes ajustes na configuração das redes com foco em estabilidade para a tarefa de diagnóstico de defeitos em chapas de aço através de imagens.

Keywords: Computer Vision; Deep Learning; Convolutional Neural Networks; Images Processing; Defect Detection; Metallurgy;

Palavras-chaves: Visão Computacional; Aprendizado Profundo; Redes Neurais Convolucionais; Processamento de Imagens; Detecção de Defeitos; Metalurgia;

1. INTRODUÇÃO

A relevância técnica e econômica do aço inox faz com que este material seja uma das principais matérias-primas da modernidade. Seu processo produtivo é especialmente delicado, o que contempla a participação de diversas organizações globalmente, sendo cada uma responsável por um grupo de processos. Por exemplo, podemos citar aquelas responsáveis pela extração de minerais em sua forma bruta, assim como operadores logísticos e fábricas. Essas são responsáveis por transformar a matéria-prima em vergalhões, tubos, bobinas e chapas: as unidades básicas para distribuição do aço inox e sua posterior disponibilização para indústrias de manufatura como a automobilística, aviação, metal-mecânica, alimentícia e tantas outras.

Durante todas as etapas do processo de fabricação de chapas de aço, diversos equipamentos interagem com os produtos, muitas vezes entrando em contato de maneira inadequada e acarretando em defeitos, avarias e consequências indesejadas. Não conformidades no produto final levam

fabricantes a tomar ações indesejáveis, como retrabalho e reparos nas peças, venda a preços menores e até mesmo o sucateamento ou descarte. Além desses ônus, os defeitos ainda podem acarretar em perdas de propriedades mecânicas dos materiais, conforme exposto por Li et al. (2018).

Ainda segundo os autores, existem fatores que dificultam a inspeção de chapas de aço em tempo real, como a velocidade elevada das linhas de produção, diversidade de defeitos e interferências não defeituosas, como manchas de óleo e poeira. Dessa forma, a proposição de um artifício para detecção automática desses defeitos pode reduzir os custos associados às inconformidades.

O termo "detecção automática" pode sugerir a presença de um sistema que classifica os defeitos a partir de um conjunto de regras definidas por especialistas. Porém essa é uma tarefa extremamente complexa, dado que a tradução dessas regras para um sistema computadorizado requer um esforço técnico considerável, uma vez que trata-se de conhecimento tácito. Ainda que possível agir dessa forma, o advento de tecnologias de *Deep Learning* - a exemplo de redes neurais convolucionais - permite com que um sistema de detecção automática seja desenvolvido sem a escrita explícita das regras que regem o fenômeno, desde que expostos à uma quantidade suficientemente grande de dados e parametrizados adequadamente.

Segundo Goodfellow et al. (2016), redes neurais convolucionais são um tipo especializado de algoritmo de aprendizado, propício para aplicações de classificação ou regressão a partir de objetos que apresentem uma topologia similar à uma malha, como séries temporais (intervalos regulares de tempo) ou imagens (malha bidimensional de *pixels*). O método têm se destacado por sua eficiência em aplicações práticas, principalmente no contexto de imagens, como iremos explorar no presente artigo.

A fabricante russa de chapas de aço Severstal captura, através de câmeras de alta frequência, imagens dos itens que produz, tendo construído um rico conjunto de dados. Em busca de eficiência em seu processo produtivo, a empresa publicou sua base em Kaggle (2019), que promove o desenvolvimento de algoritmos de Inteligência Artificial em um ambiente de competição. Além do propósito original da plataforma de promover a solução de um problema diretamente associado à indústria, a disponibilização de seus dados favorece o desenvolvimento acadêmico, dado que o acesso à esse tipo de informação geralmente está vinculado à iniciativa privada. Ainda que os autores do presente artigo não tenham participado da competição, o conjunto de dados fora utilizado como cenário para o desenvolvimento acadêmico de conceitos de redes neurais. Dessa forma, destacamos que o cerne deste trabalho é distinto da proposta original solicitada na plataforma *Kaggle*.

1.1 Objetivo

O trabalho possui caráter exploratório e comparativo, focado na tarefa de classificação de imagens de chapas de aço, diagnosticando à qual tipo de defeito (rótulo) estas estão associadas. Além da classificação, o presente trabalho busca a análise de parâmetros que tragam estabilidade ao algoritmo. Assim, o presente artigo aprofunda-se no estudo de técnicas de regularização que favoreçam a confiabilidade na performance, generalização e robustez desse tipo de modelo.

2. FAMILIARIZAÇÃO COM O CONJUNTO DE DADOS

2.1 Análise Preliminar do Conjunto de Dados

O conjunto de dados apresenta imagens de cinco categorias distintas, sendo uma delas para chapas sem avarias e quatro para chapas defeituosas. Inclusive, chapas podem apresentar defeitos de mais de uma categoria, porém em regiões diferentes da peça. Uma vez que estamos buscando associar apenas um rótulo às chapas de acordo com os defeitos que elas apresentam, imagens com mais de um defeito foram excluídas da análise e foram removidas do escopo deste trabalho. Ainda que não seja o foco deste trabalho, comenta-se que uma possível abordagem para esse tipo de problema seria treinar múltiplas redes neurais, cada uma especializada em um tipo único de defeito. Uma vez que o objetivo aqui é a exploração de técnicas para regularização e estabilidade do modelo, o treinamento e a parametrização de diversas redes neurais se inviabiliza dado o recurso computacional disponível.

A complexidade do problema é destacada ao analisar-se mais de uma amostra da classe sem defeitos. Há imagens que apresentam aparentes irregularidades superficiais, mas ainda assim são rotuladas como conformes. Isso pode significar que manchas em seu acabamento estão de acordo com os padrões de qualidade esperados pelo fabricante e podem ser oriundas de diferenças de coloração ou manchas no material, sem prejuízo para o usuário final. Partindo desse pressuposto, é possível inferir visualmente que as amostras consideradas defeituosas possuem marcas mais profundas no material. Porém, o relevo pode não ser o único atributo determinante do defeito das chapas, dada a existência de chapas com texturizadas no grupo de imagens sem defeitos.

Amostras de imagens para cada uma das classes e respectivas hipóteses dos autores para suas causas são expostas em Fig. 1 a 6.

Fig. 1. Chapa pertencente à classe 0 (sem avarias).

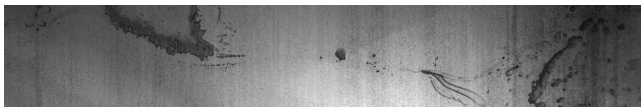

Fig. 2. Chapa pertencente à classe 0 (sem avarias). Apresenta superfície não uniforme.

Fig. 3. Chapa pertencente à classe 1: Perfurações na superfície das chapas. Possivelmente corrosão alveolar.

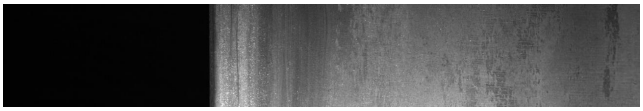

Fig. 4. Chapa pertencente à classe 2: Marcas verticais nas bordas das chapas. Possivelmente foram resultado de áreas de contato com equipamentos para manipulação das chapas, como ganchos para içamento.

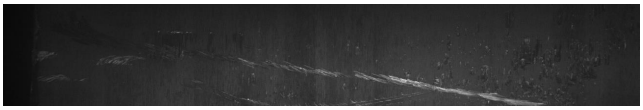

Fig. 5. Chapa pertencente à classe 3: Marcas com orientações e formatos variados. Ao analisar visualmente, aparentam ser consequências de impactos ou arranhões da manipulação das chapas.

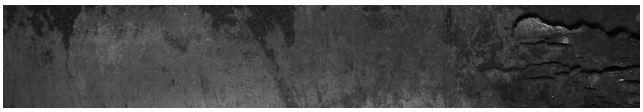

Fig. 6. Chapa pertencente à classe 4: Partes profundamente descascadas ou com irregularidades mais proeminentes na superfície das chapas.

2.2 Distribuição do Conjunto de Dados

É possível visualizar que não há uma quantidade igual de imagens para cada classe, vide Fig. 7. Uma vez que o conjunto de dados tem um tamanho considerável, infere-se que a diferença é intrínseca ao processo de fabricação das chapas. Conjuntos de dados dessa natureza são chamados de desbalanceados. Em uma população de 12568 imagens, temos aproximadamente 46,96%, 7,13% 1,67%, 40,12% e 4,10% respectivamente para as classes 0, 1, 2, 3 e 4.

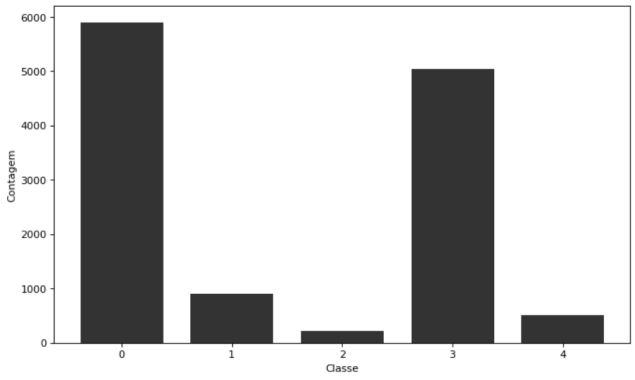

Fig. 7. Distribuição de classes no conjunto de dados

2.3 Aspectos Relevantes das Amostras

As imagens das chapas foram registradas a partir de câmeras de alta frequência e resolução, com dimensões de 1600 x 256 *pixels*. Cores não apresentam um fator relevante para a detecção dos defeitos, principalmente devido à coloração natural das chapas. Assim, a intensidade do brilho de cada pixel e de seus respectivos vizinhos são as características a serem inseridas nas entradas da rede neural proposta. Dessa forma, trabalharemos com as imagens apenas em um canal, referente à escala de cinza.

Além do ajuste de cores, normalizamos as imagens para que os *pixels* tivessem valores entre 0 (*pixel* mais escuro) e 1 (*pixel* mais claro). Reduzimos também o tamanho das imagens para as dimensões de 160 x 25 *pixels*. Dessa forma, é possível executar o treinamento em lotes (*batches*) de imagens, tornando a tarefa da mais efetiva em termos de tempo e recursos computacionais. Ainda que informações da imagem original sejam perdidas nessa redução, o ganho de tempo no treinamento da rede neural supera os aspectos negativos dessa decisão.

3. CONFIGURAÇÃO INICIAL DA REDE NEURAL

Existem diversos parâmetros para a configuração de redes neurais, a exemplo de números de camadas, neurônios, funções de ativação, filtros de convolução e taxa de aprendizado. Buscar ajustes de parâmetros que forneçam o melhor resultado em uma rede neural é, em geral, um processo laborioso e exaustivo. Sem uma comunhão científica sobre como proceder com a busca, o empirismo é amplamente adotado, até mesmo por especialistas na área, principalmente devido à baixa explicabilidade de previsões de redes neurais. Ainda que melhores práticas para a parametrização sejam discutidas, nota-se que não há uma solução aplicável à todos os casos. As particularidades do problema e distribuição dos seus dados devem ser analisadas individualmente.

Em consideração ao objetivo do trabalho em explorar comparativamente técnicas voltadas para ganho em generalização e regularização do modelo, os esforços despendidos para o ajuste de parâmetros básicos foram executados através da busca exaustiva com intervalos limitados *Grid Search*. De acordo com Geron (2019), define-se um conjunto finito de valores a serem explorados para cada parâmetro. Em seguida, o modelo é treinado em cada uma das combinações de parâmetros, visando a melhor performance nos dados de teste. O menor e o maior valores do conjunto são definidos de maneira conservadora, baseada na experiência prévia em experimentos similares.

Aqui, o *Grid Search* foi aplicado para a quantidade de camadas ocultas na rede e a quantidade de neurônios de cada camada, propondo uma configuração a ser utilizada como referência de performance (*Benchmark*). A rede pode ser dividida em dois grandes blocos, sendo o primeiro constituído por camadas convolucionais associadas à camadas de *Pooling* e o último por camadas densamente conectadas.

O primeiro bloco é composto de duas camadas com a seguinte configuração: convolução, ativação *ReLu* e *Max Pooling*. Na convolução foram implementados 64 filtros com *Kernels* de dimensão de 3 x 3 por camada. O *Max-Pooling* foi executado na dimensão de 2 x 2 por camada, conforme recomendado por Boureau et al. (2010).

O segundo bloco é composto por três camadas densamente conectadas com ativações *ReLu* e uma quarta com ativação *Softmax* para a saída. A função de ativação *Softmax* na camada de saída permite com que o algoritmo atribua uma probabilidade para cada uma das classes analisadas no problema. A soma das probabilidades de saída de todas as classes é 1 (ou 100%). Na implementação da rede neural (através do pacote *TensorFlow*), o neurônio com maior probabilidade na última camada é responsável

pela definição da classe imagem de entrada, ainda que sua probabilidade não seja igual a 100%.

A função de perda foi a entropia cruzada categórica, indicada para classificação multi-classes, segundo Chollet (2018). Essa função minimiza a distância entre duas distribuições de probabilidade. No caso de redes neurais, essas são a distribuição de previsões fornecidas pela rede e a verdadeira distribuição dos rótulos. Assim, a rede entregará um resultado próximo aos rótulos originais. A minimização se dará através do processo de *backpropagation*, em que o erro das previsões feitas pela rede ao longo das épocas de treinamento serão revertidos na melhoria dos ajustes de pesos e vieses das camadas da rede neural, propagados de trás para frente, vide Rumelhart et al. (1986).

A escolha do otimizador foi o Adam (termo derivado de *"Adaptative Moments"*). Este otimizador geralmente apresenta grande robustez em relação aos outros parâmetros da rede, se tornando a principal escolha para esse parâmetro na maioria das redes neurais atualmente, conforme Goodfellow et al. (2016). Além desses parâmetros, foi utilizada também a função *Early Stopping"*. A técnica permite com que o treinamento seja encerrado após a função de perda atingir seu mínimo. Em seguida, o melhor modelo ao longo das épocas é retornado. Uma vez a redução da função de perda é irregular e pode oscilar na medida em que é minimizado, o parâmetro *patience* foi ajustado para 15 épocas. Assim, o algoritmo não encerra o treinamento imediatamente e permite que o melhor modelo seja retornado.

A taxa de aprendizado utilizada foi a padrão da biblioteca *TensorFlow*, de 0.01. Demais parâmetros ajustáveis da rede não foram modificados nesse estágio. Em seguida, prosseguimos com o experimento, exploração e análise das capacidades da rede desenvolvida.

4. EXPERIMENTOS

4.1 Padronização dos Experimentos e Métricas de Avaliação

Para que o estudo permita a realização de uma análise comparativa, é necessário que sua execução ocorra de forma padronizada e seus resultados sejam mensurados quantitativamente. Buscou-se significância estatística nos experimentos devido à dois fatores. Primeiramente, redes neurais são algoritmos que apresentam elevada influência de aleatoriedade, dado que seus pesos são inicializados randomicamente. Em segundo lugar, os treinamentos das redes neurais foram paralelizados através de *Graphics Processing Units* (GPUs). Nesse processamento, não é possível garantir que todas as operações irão ocorrer da mesma ordem em cada treinamento, ao contrário do treinamento realizado via *Central Processing Unit* (CPU). O treinamento em CPUs é imensamente mais devagar que aquele realizado em GPUs, o que justifica a decisão.

Para a obtenção da significância estatística nos experimentos, os processos de partição do conjunto de dados em treino e teste, aprendizado da rede e validação foi realizado 100 vezes em cada experimento. A separação do conjunto de dados foi realizada de maneira aleatória e independente em cada experimento, nas proporções de 80% e 20% para os dados de treino e teste, respectivamente.

A análise das métricas de acurácia, precisão, revocação e *F1 Score* foi realizada após a obtenção da média das iterações. Além da média, a estabilidade do modelo foi analisada através do desvio padrão de cada uma das métricas.

Outra consideração importante para a padronização dos experimentos é a determinação do ponto de sobreajuste. Segundo James et al. (2013), este se dá a partir do ponto em que o modelo aprende erros e ruído ao invés das características dos dados. Nesse ponto, a função de perda aumenta seu valor. Assim, decidiu-se a utilização da técnica de *Early Stopping* para encerramento de cada treinamento e determinação do melhor modelo, o qual foi utilizado para a etapa de validação. Foram executadas 30 épocas de treinamento. Cada época consiste na apresentação integral de todos dados de treinamento para a rede neural.

Nas próximas subseções descreve-se cada um dos experimentos e na seção de resultados estes serão comentados.

4.2 Benchmark

O primeiro modelo treinado foi parametrizado exatamente conforme o apresentado na seção 4. Na Fig. 8, é possível visualizar em destaque a curva de perda média do modelo e as curvas de cada um dos treinamentos ao fundo.

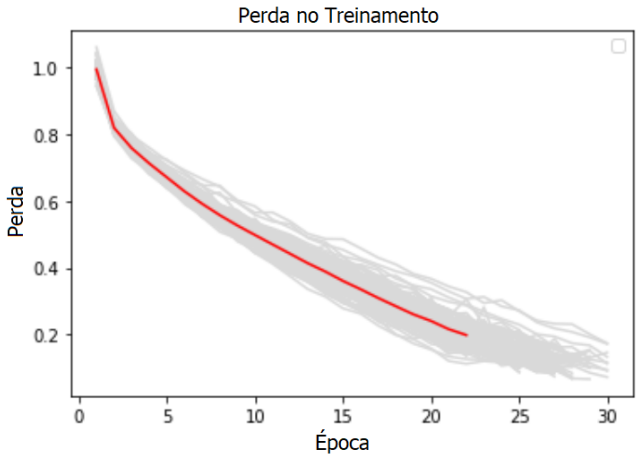

Fig. 8. Benchmark: Curvas de Perda no Treinamento

Na medida que o treinamento avança, é possível visualizar que a função de perda tende à minimização. Porém, só possível avaliar se a rede realmente é capaz de generalizar a partir da sua aplicação em dados novos, como o subconjunto de teste. Neste caso, é possível visualizar na Fig. 9 significativa perda na performance à partir da 10ª época, fenômeno que indica a presença de sobreajuste e perda na capacidade de generalização. Outro fator que se destaca no gráfico é a instabilidade da função de perda, ressaltando a necessidade de ajustes nesse sentido.

O comportamento de sobreajuste também pode ser visualizado na Fig. 10 ao sobrepormos as curvas de média, na medida que as curvas se distanciam. Ainda que esse indesejável comportamento foi percebido, a utilização técnica de *Early Stopping* permitiu com que o os modelos com menor perda fossem utilizados para a apuração de resultados em dados novos.

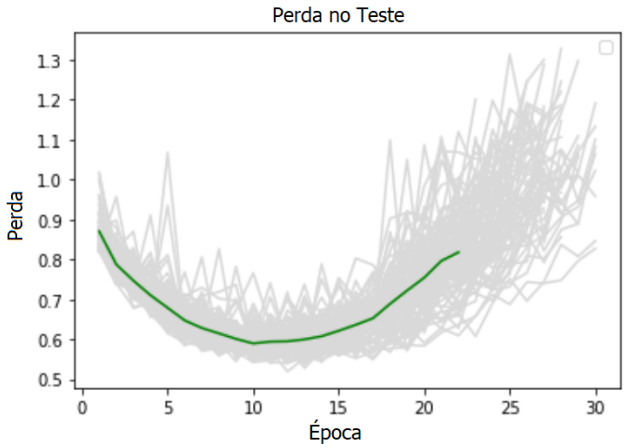

Fig. 9. Benchmark: Curvas de Perda no Teste

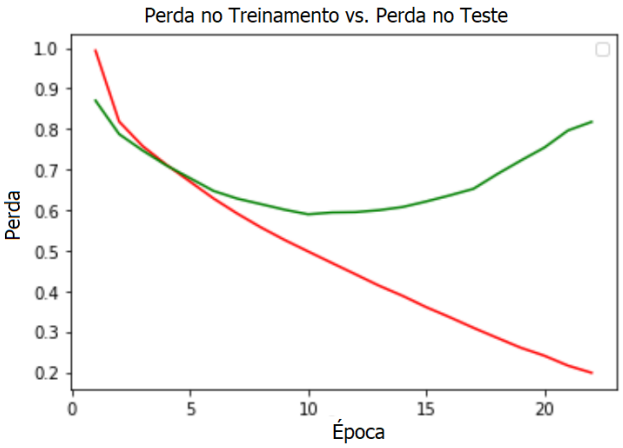

Fig. 10. Benchmark: Curvas sobrepostas e sobreajuste

4.3 Data Augmentation

Uma abordagem para a melhoria de performance em modelos de aprendizado de máquina é o aumento da qualidade e/ou da quantidade dos dados. A técnica de *Data Augmentation* é uma ferramenta utilizada com esse fim principalmente no contexto de imagens. A técnica consiste na aplicação de transformações nos dados disponíveis inicialmente, promovendo a criação de novas amostras realistas. Em geral, essas transformações consistem na rotação e distorções das imagens. A aplicação de técnica força o modelo a ser mais tolerante a variações na posição, orientação e tamanho de objetos nas imagens.

Optou-se por utilizar a técnica com o intuito de criar novas amostras para as classes com chapas defeituosas, uma vez que essas são menos frequentes. Com essa abordagem foi possível reduzir, também, o desbalanceamento do conjunto de dados.

Uma vez que as imagens foram capturadas de maneira padronizada, é recomendado que a aplicação de rotações e inversões mantenham a uniformidade do conjunto de dados. Caso contrário, as novas amostras não serão representativas do fenômeno e a performance no conjunto de teste não será melhorada. A partir disso, realizou-se apenas rotações e inversões vertical e horizontalmente, limitando-se à adição de três novas amostras para cada imagem e respeitando a quantidade máxima de imagens por classe, para não provocar novo desbalanceamento. O conjunto de dados ampliou em 45,57%. Em Fig. 11, apresentamos a diferença das distribuições antes e depois da expansão:

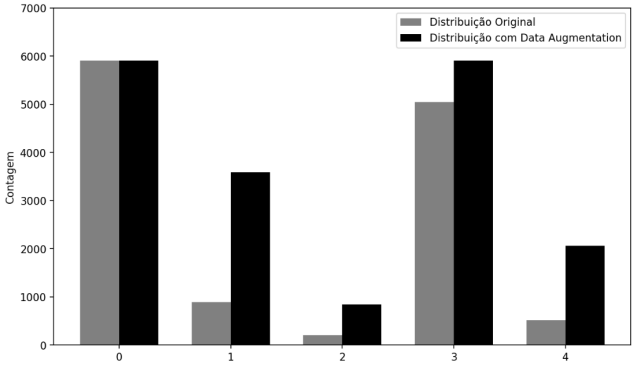

Fig. 11. *Data Augmentation*: Distribuição das Classes

A partir da análise da Fig. 12, é possível atestar que a adição de novos dados não foi suficiente para a melhoria em relação ao sobreajuste. Acredita-se que, para processamento da nova quantidade de dados proposta, uma rede neural com topologia distinta deverá ser implementada. Dessa forma, uma vez que o presente estudo tem caráter comparativo, uma rede configurada de maneira diferente seria fundamentalmente diferente, invalidando comparações. Ainda que na atual conjuntura a técnica de *Data Augmentation* não tenha apresentado melhorias, sua presença na literatura estimula a sua exploração em trabalhos futuros.

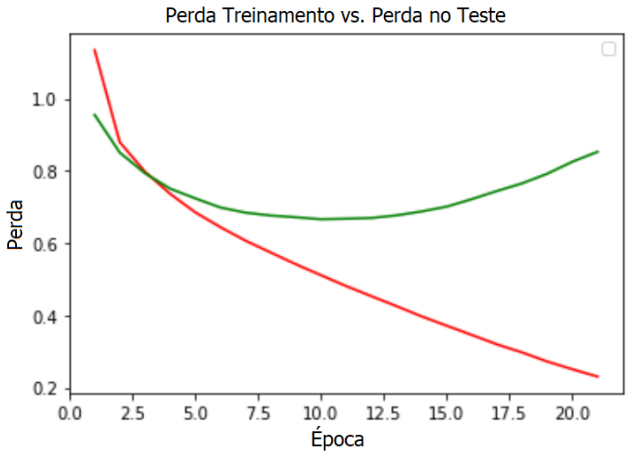

Fig. 12. *Data Augmentation*: Curvas sobrepostas

4.4 Ajuste da Taxa de Aprendizado

A taxa de aprendizado é parte importante do processo de otimização da função de perda. Seu ajuste apresenta uma dualidade: se muito grande, favorece a velocidade do treinamento mas pode fazer com que o modelo não tenha convergência; se muito pequena, terá mais chances de convergir, mas o treinamento pode se tornar excessivamente longo. Para mitigar o problema da instabilidade, propõe a utilização de um agendamento da taxa de aprendizado, de maneira com que ela reduza ao longo do treinamento.

O trabalho de Senior et al. (2013) comparou a performance de algumas técnicas de agendamento de taxas de aprendizado. No contexto do artigo, o agendamento exponencial apresentou rápida convergência, facilidade de implementação e parametrização. Dessa forma, optamos por selecionar o método em nosso trabalho. Segundo Senior et al. (2013), sua fórmula pode ser dada por (1).

$$\eta(t) = \eta_0 * 0,1^{(t/s)} \tag{1}$$

Sendo η a taxa de aprendizado, t a época de treinamento e s rapidez do decaimento. Não existe na literatura uma regra para a definição da taxa de aprendizado e, consequentemente, o mesmo vale para a definição do parâmetro s. Após iterações, obtivemos resultados satisfatórios com $\eta = 0{,}001$ e $s = 15$. A influência de s pode ser observados na Fig. 13.

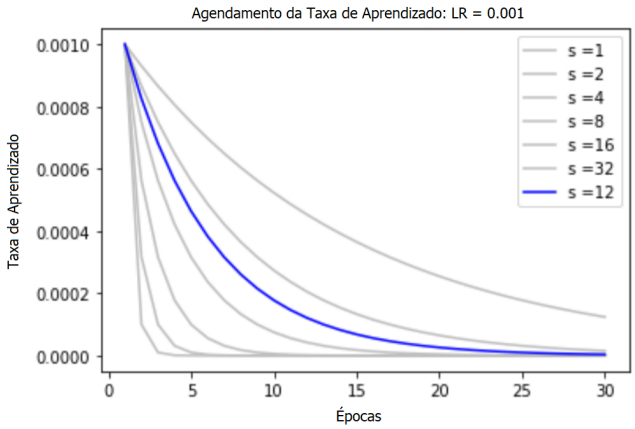

Fig. 13. Comportamento da taxa de aprendizado em relação à variações em s.

A técnica favoreceu a performance do modelo. Dados os resultados positivos da sua aplicação e devido ao fato de que esta não é uma técnica dedicada exclusivamente à regularização, ela foi adotada para os outros experimentos realizados com os mesmos ajustes dessa seção.

4.5 Class Weights

Ainda que a aplicação do agendamento tenha sido efetiva, a instabilidade do modelo não foi reduzida inteiramente. Assim, outro recurso foi aplicado como tentativa para a mitigação de efeitos indesejados. O parâmetro *Class Weight*, disponível na biblioteca *TensorFlow* de Abadi et al. (2015), em que as redes foram implementadas, permite com que classes com diferentes quantidades de amostras sejam ponderadas durante o processo de treinamento. Classes menos expressivas recebem um peso maior que aquelas preponderantes no cálculo de perda. Sua aplicação apresentou resultados favoráveis, dado o expressivo desbalanço do conjunto de dados.

4.6 Label Smoothing

A técnica de *Label Smoothing* consiste não na alteração de características da rede neural, mas sim na características dos rótulos, assim como no processo de *Data Augmentation*. Destaca-se nesse ponto que os rótulos das imagens são constituídos por vetores $\delta_{k,y}$, sendo y o elemento da classe verdadeira e k o comprimento do vetor. O valor de k é de dimensão igual à última camada da rede neural e a soma de seus elementos resulta em 1 (ou 100%). Para que a soma aconteça dessa maneira, a função de ativação *Softmax* deve ser utilizada. Essa é uma função não linear com limites superior e inferior tendendo a 1 e 0, respectivamente. Por esse comportamento da *Softmax*, a probabilidade entregue pela rede neural nunca irá atingir 100% de certeza para uma única classe (ou 0% para as demais), a menos que o valor de entrada na última camada seja igual a infinito, o que nunca irá acontecer na prática.

Na concepção dos rótulos, é de praxe que a classe positiva assuma o valor de 1 no vetor, enquanto as classes negativas sejam iguais à 0. Porém, essa situação só ocorrerá hipoteticamente, dado que os valores de entrada na última camada não atingirão infinito. Na prática, a rede neural será estimulada a aumentar os pesos das conexões entre os neurônios indiscriminadamente, segundo Goodfellow et al. (2016). Dessa forma, mesmo que a classe com o maior valor no vetor de saída seja a classe verdadeira, a rede continuará a aumentar seus pesos, o que reduz as capacidades de generalização da rede.

Ao aplicar o *Label Smoothing*, força-se que os rótulos reduzam seus valores para a classe positiva e aumentem os valores para as classes negativas. Assim, o aumento dos pesos nas conexões entre os neurônios será encerrado limitado, já que os rótulos e as previsões irão atingir o mesmo valor. Dessa forma, pesos excessivamente grandes não irão fazer parte da rede neural e as previsões serão melhores. Segundo Szegedy et al. (2015), o novo vetor $q'(k)$ do rótulo $\delta_{k,y}$ é dado por (2).

$$q'(k) = (1 - \epsilon)\delta_{k,y} + \epsilon/k \tag{2}$$

A partir de (2), a classe positiva teve seu rótulo ajustado de 100% para 92% e as classes negativas foram alteradas de 0% para 2%. A técnica apresentou resultados agradáveis, principalmente no que se tange à estabilidade do modelo.

4.7 Dropout

Uma das técnicas de regularização mais populares é o *Dropout*. É baseado em *Ensembles*, que consistem na combinação de modelos de aprendizado de máquina. Ao combinar modelos diferentes, o coletivo de suas previsões pode ser melhor do que aquelas de cada modelo individualmente. No contexto de redes neurais, os modelos individuais são obtidos a partir do desligamento de neurônios aleatórios durante o treinamento. Isso fará com que a rede se torne menos dependente dos neurônios apagados e amplie sua capacidade de generalização com menos recursos. Ao final das épocas de treinamento, todos os neurônios são ativados e, como um *Ensemble*, a rede tende a performar melhor do que aquelas com neurônios desabilitados, conforme Srivastava et al. (2014).

No presente trabalho, explorou-se redes com *Dropout* de 10% e 45% dos neurônios. O ajuste menor performou significativamente melhor do que o mais agressivo, ainda que seu valor tenha sido inferior ao recomendado para redes convolucionais em Geron (2019). Acredita-se que o

resultado se deve ao tamanho da rede neural desenvolvida pelos autores, com quantidade de neurônios relativamente pequena. Ainda que na atual conjuntura o ajuste de *Dropout* com taxas mais elevadas não tenha apresentado melhorias expressivas, sua divulgação na literatura encoraja a sua exploração em redes maiores.

4.8 Regularização L2

A Regularização L2 (também apresentada como *Ridge Regression*) consiste na aplicação de uma penalidade ao modelo. Segundo Geron (2019), é feito ao adicionar o termo $\alpha \sum_{i=1}^{n} (\theta_i)^2$ na função objetivo. No termo, α é o coeficiente de regularização e θ representa valores dos pesos das conexões da rede neural. Assim, a regularização promove com que os pesos do modelo assumam pequenos valores na função de custo, reduzindo a sua não-linearidade e favorecendo a generalização perante a apresentação de novos dados, conforme proposto por Hinton et al. (2012). Aqui, o parâmetro de regularização α foi ajustado para 0.001 e sua aplicação contribuiu positivamente para a performance do algoritmo.

5. ANÁLISE DE RESULTADOS

A métrica de análise para classificação deve ser atrelada ao contexto do problema a ser resolvido ou ao indicador de negócios a qual está submetido. Dessa forma, dado que a experiência dos autores é limitada no domínio do problema, definir precisamente qual a métrica ideal foge do escopo do trabalho. Assim, apresenta-se em seguida a performance dos experimentos em relação às métricas de acurácia, precisão, revocação e *F1 Score*.

5.1 Acurácia

A métrica de acurácia, apesar de ser a mais simples de compreender (porcentagem de previsões positivas), é muita das vezes inadequada para a problemas desbalanceados, dado que as classes podem representar grande parcela do conjunto de dados e, de certa forma, prejudicar a interpretabilidade ao apresentar resultados aparentemente bons. Conforme sugerido na seção 3.1, o desenvolvimento de modelos especialistas em cada uma das classes poderia se beneficiar de métricas como a acurácia se analisados individualmente. De qualquer maneira, os experimentos de Agendamento da Taxa de Aprendizado, *Class Weights*, *Label Smoothing* e *Dropout* ajustado para 10% apresentaram valores melhores do que o *Benchmark*, vide Tabela 1.

Tabela 1. Acurácia.

Experimento	μ	σ
Benchmark	0.79081	0.00881
Data Augmentation	0.76060	0.00804
Agendamento da Taxa de Aprendizado	0.80389	0.00646
Class Weights	0.79934	0.00606
Label Smoothing	0.80813	0.00633
Dropout 10%	0.80548	0.00839
Dropout 45%	0.76316	0.01233
L2	0.80091	0.00583

5.2 Precisão

Assim como na métrica de acurácia, a técnica de *Label Smoothing* se destacou no quesito de precisão. A métrica pode ser compreendida como a razão de verdadeiros positivos pela quantidade de previsões positivas (verdadeiras e falsas). Apesar da melhor performance média de *Label Smoothing*, a rede neural com *Dropout* de 10% apresentou o valor máximo de precisão perante os demais experimentos, ainda que sua média tenha sido menor. Isso mostra menor estabilidade desse modelo, ainda que este seja promissor. Apresentamos os resultados na Tabela 2.

Tabela 2. Precisão.

Experimento	μ	σ
Benchmark	0.78411	0.01046
Data Augmentation	0.76120	0.00791
Agendamento da Taxa de Aprendizado	0.79552	0.00737
Class Weights	0.79728	0.00651
Label Smoothing	0.79945	0.00784
Dropout 10%	0.79701	0.01073
Dropout 45%	0.70293	0.03069
L2	0.79205	0.00804

5.3 Revocação

A métrica de Revocação ou *Recall* consiste na taxa de verdadeiros positivos previstos em relação à soma da quantidade de verdadeiros positivos e falsos negativos. Novamente, a melhor performance foi entregue pelo modelo com *Label Smoothing*. Resultados na Tabela 3.

Tabela 3. Revocação.

Experimento	μ	σ
Benchmark	0.79081	0.00881
Data Augmentation	0.76060	0.00804
Agendamento da Taxa de Aprendizado	0.80389	0.00646
Class Weights	0.79934	0.00606
Label Smoothing	0.80813	0.00633
Dropout 10%	0.80548	0.00839
Dropout 45%	0.76316	0.01233
L2	0.80091	0.00583

5.4 F1 Score

A métrica de *F1 Score* promove uma maneira útil de comparar classificadores, dado que ela consiste na média harmônica entre precisão e revocação. Dessa forma, ela irá apresentar seus maiores valores quando precisão e revocação apresentarem resultados próximos, o que foi o caso da técnica de *Class Weights*, vide Tabela 4.

Tabela 4. F1 Score.

Experimento	μ	σ
Benchmark	0.78091	0.01061
Data Augmentation	0.75864	0.00840
Agendamento da Taxa de Aprendizado	0.79449	0.00754
Class Weights	0.79767	0.00635
Label Smoothing	0.79644	0.00776
Dropout 10%	0.79343	0.01074
Dropout 45%	0.72893	0.02152
L2	0.78844	0.00706

5.5 Combinação de Técnicas

Uma vez que as redes com agendamento da taxa de aprendizado, *Class Weights*, *Label Smoothing* e *Dropout* mais conservador superaram a performance do *Benchmark* em todas as métricas e essas não são mutuamente exclusivas, uma rede com sua combinação foi treinada nos mesmos moldes do experimento. As Tabelas 5 a 8 apresentam seus resultados em relação ao *Benchmark*.

Tabela 5. Acurácia

Experimento	μ	σ
Benchmark	0.79081	0.00881
Combinação de Técnicas	0.81015	0.00718

Tabela 6. Precisão

Experimento	μ	σ
Benchmark	0.78411	0.01046
Combinação de Técnicas	0.79734	0.01179

Tabela 7. Revocação

Experimento	μ	σ
Benchmark	0.79081	0.00881
Combinação de Técnicas	0.81015	0.00718

Tabela 8. F1 Score

Experimento	μ	σ
Benchmark	0.78091	0.01061
Combinação de Técnicas	0.79456	0.00895

6. CONCLUSÕES

O espaço de soluções e parâmetros para uma rede neural é de uma dimensão expressiva. Dessa forma, sua exploração depende do custo de computação disponível. Dessa maneira, técnicas consolidadas na literatura foram priorizadas. Ainda assim, as condições particulares do estudo de caso fortaleceram que não há solução universal para a melhoria de algoritmos de aprendizado. Em nosso artigo, a técnica de *Label Smoothing* se mostrou especialmente efetiva em várias métricas. Dessa forma, a sua aplicação em outros contextos é estimulada.

Técnicas como *Data Augmentation* e ajustes mais agressivos de *Dropout* não performaram como esperado. Concluise que essas possuem contribuições maiores em outras topologias de redes neurais, como redes mais profundas. Agendar o decaimento da taxa de aprendizado também trouxe contribuições interessantes para a estabilidade do modelo, mas pode levar o modelo à uma convergência prematura.

Finalmente, a participação de um profissional especialista no domínio de metalurgia poderia trazer informações que enriqueceriam o processo, a exemplo da indicação da melhor métrica de performance para o problema.

Em trabalhos futuros pretende-se, além de classificar, localizar os defeitos nas imagens, destacando sua posição através de *bounding boxes* ou mesmo contornando sua forma precisamente, processo chamado de segmentação.

7. AGRADECIMENTOS

Agradecimento ao professor Dr. Gustavo Luís Soares, pelo incentivo à submissão desse trabalho.

REFERÊNCIAS

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Boureau, Y.L., Ponce, J., and LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 111–118. Omnipress, Madison, WI, USA.

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications Co, 1st edition.

Geron, A. (2019). *Hands-on Machine Learning with Scikit–Learn, Keras and TensorFlow*. O'Reilly Media, 2nd edition.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, 1st edition.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580. URL http://arxiv.org/abs/1207.0580. Cite arxiv:1207.0580.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. URL https://faculty.marshall.usc.edu/gareth-james/ISL/.

Kaggle (2019). Severstal: Steel defect detection. URL https://www.kaggle.com/c/severstal-steel-defect-detection.

Li, J., Su, Z., Geng, J., and Yin, Y. (2018). Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC-PapersOnLine*, 51, 76–81. doi:10.1016/j.ifacol.2018.09.412.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning Representations by Back-propagating Errors. *Nature*, 323(6088), 533–536. doi:10.1038/323533a0. URL http://www.nature.com/articles/323533a0.

Senior, A., Heigold, G., Ranzato, M., and Yang, K. (2013). An empirical study of learning rates in deep neural networks for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver, CA.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958. URL http://jmlr.org/papers/v15/srivastava14a.html.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567. URL http://arxiv.org/abs/1512.00567.