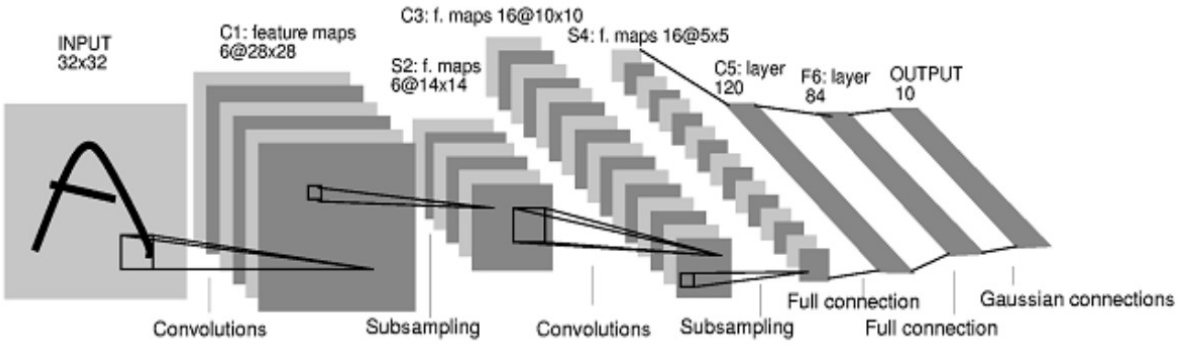
INTRODUÇÃO

Neste laboratório vamos trabalhar com data Augmentation que será implementado para aumentar o tamanho das amostras e algumas redes convolucionais.

Para a rede LeNet 5 foi utilizado os parâmetros da figura dada em aula.

Esse conceito foi popularizado por Yann LeCun em 1998, trata-se de uma CNN com 7 camadas utilizada na época para reconhecer dígitos manuscritos quando desenvolvida teve baixa popularização devido ao seu auto custo computacional.



Na segunda CNN (aleatória) conforme solicitada no trabalho foi utilizado parâmetros aleatórios (próximo ao que foi apresentado em aula), conforme abaixo.

```
model = Sequential()
model.add(Conv2D(32,(8, 8), strides=(1, 1), activation='relu', padding='valid',
input_shape=input_shape))
model.add(Conv2D(128, (2, 2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.20))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(num_classes, activation='softmax'))
```

OBJETIVOS

Conforme relatado no moodle, o objetivo desse laboratório é analisar uma base de dados com imagens manuscritas dos doze meses do ano através das práticas de Deep Learning apresentada em aula.

Sobre essa base de dados deveremos utilizar funções de data augmentation, com o intuito de aumentar a base de treinamento, criando uma quantidade significante de amostras para verificar o comportamento das duas CNNs.

Conforme citado anteriormente deverá ser implementado ainda duas rede neurais a LeNet 5 [Yann LeCun (1998)] e uma CNN a escolha.

E por fim relatar os acontecimentos com e sem data augmentation, comparando seus desempenhos apresentando o gráfico da acurácia e a matriz de confusão e ainda os experimentos deverão ser repetidos em outro classificador como o SVM.

O objetivo do laboratório é analisar a base de dados fornecida, comparando os desempenhos como tempo de execução, acurácia para identificar assim o mais rápido e também o que tem melhor desempenho, além de verificar ainda qual o tamanho que a base de treinamento deixa de ser relevante, e por fim analisar as matrizes de confusão para verificar os erros de cada classificador.

METODOLOGIA

Os materiais utilizados foram o Google colab, scikit learn e o libre office calc, com base nas aulas e laboratório anterior foram criados alguns algoritmos. Com isso foi possível efetuar as análises conforme orientado no documento – CNN.

Conforme solicitado no item 1 do documento, foi implementado dois algoritmos de data augmentation, um deles trata-se de pegar algumas fontes (todas retiradas do site: https://fonts.google.com) que se assemelham a escrita manual e assim gerar as imagens (.jpg), com os meses do ano, com isso a base de treinamento foi aumentada com novas imagens.

O segundo algoritmo pega as imagens existentes e a partir delas cria novas imagens alterando a rotação o ruido entre outros atributos da imagem para ampliar a base de dados.

Como base para as CNN's foi utilizado os algoritmos dados em aula, a partir dele feito as alterações para conseguirmos atingir o LeNet 5 e uma CNN aleatória.

Além disso foi feito ainda uma algoritmo com o SVM para analise e comparação, e repetição dos experimentos com as redes pré trienadas Transfer Learning e Fine-Tuning.

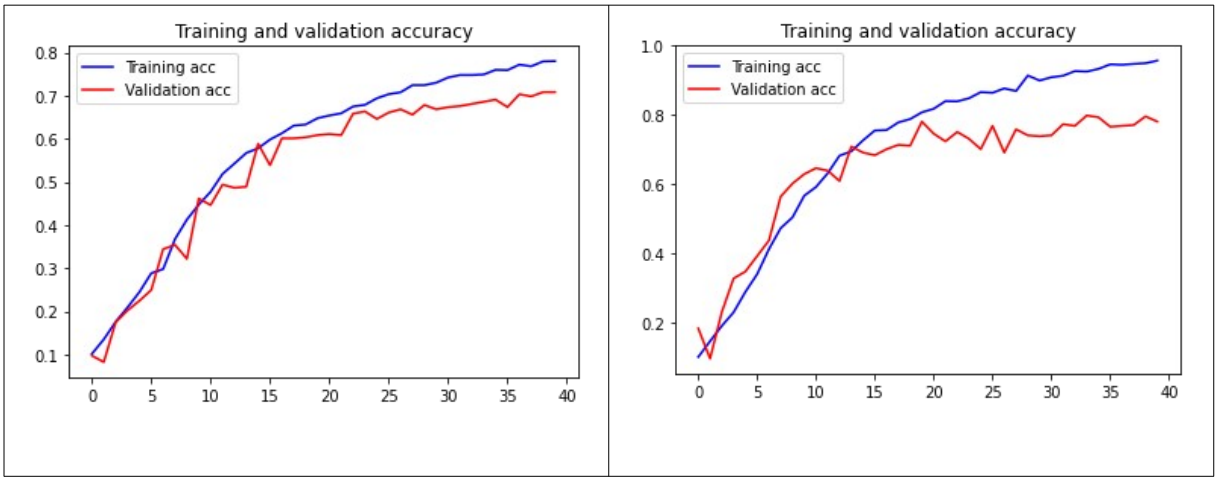
RESULTADOS E DISCUSSÃO

Foram feitos vários testes, alterando as dimensões, alterando a quantidade de épocas, entre outras alterações. Cada teste gerou uma acurácia, para apresentar os resultados foi escolhido apenas um, com características semelhantes em todas as CNN's e feito as comparações, essa maneira possibilitou um trabalho não muito extenso e também mostrou todas as comparações exigidas no documento.

O exemplo escolhido foi analisado com 40 épocas, dimensão de 64x64.

Inicialmente vamos comparar todos os exemplos sem data Augmentation:

LeNet 5	**Outra CNN**
accuracy: 0.70822	accuracy: 0.78054
Matriz de confusão	**Matriz de confusão**
[26 3 1 1 0 0 3 0 0 0 4 1] [5 19 1 0 4 0 0 1 1 1 0 0] [1 0 33 0 0 0 1 0 1 0 0 0] [2 1 1 21 8 0 0 0 3 2 0 1] [0 0 0 7 29 0 0 0 1 1 0 0] [5 1 0 0 0 18 1 2 1 0 0 1] [3 0 1 0 0 1 24 0 1 0 1 1] [0 0 0 0 1 0 0 21 0 0 1 5] [0 0 0 2 1 0 0 0 21 5 1 1] [0 0 0 1 0 0 0 0 1 28 0 0] [0 2 0 1 2 0 0 0 3 0 26 0] [0 0 0 1 0 1 0 3 2 2 6 18]	[27 4 0 0 0 5 2 0 0 0 1 0] [9 20 0 1 0 0 0 1 0 0 1 0] [0 0 35 0 0 0 0 0 1 0 0 0] [0 0 1 30 4 0 0 0 1 1 2 0] [1 0 0 2 35 0 0 0 0 0 0 0] [4 0 1 0 0 21 3 0 0 0 0 0] [4 1 1 0 0 2 23 0 0 1 0 0] [0 0 1 1 0 0 0 22 0 0 1 3] [2 0 0 2 0 1 0 0 20 4 2 0] [0 0 1 0 0 0 0 0 0 29 0 0] [0 1 0 0 1 0 0 0 2 1 28 1] [0 0 0 0 0 1 0 2 2 2 3 23]



O resultado de ambas as redes analisadas sem data augmentation foram bem semelhantes com pequena vantagem para a segunda CNN, e conforme observado nas matrizes de confusão os erros foram bem semelhantes, com as principais confusões nos meses de janeiro e fevereiro, junho e julho e dezembro e novembro.

Para apresentar os resultados com data augmentation será utilizado o mesmo raciocínio anterior, 40 épocas, dimensão 64 x 64 e rodamos com o mesmo algoritmo porém com uma base de treinamento aumentada.

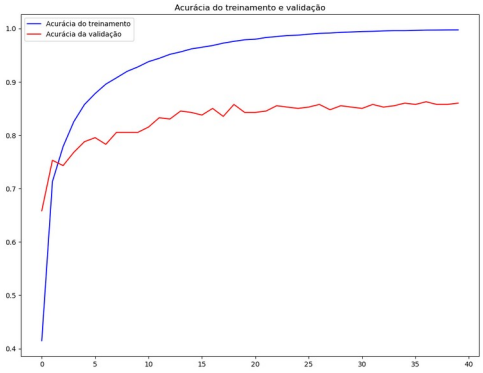
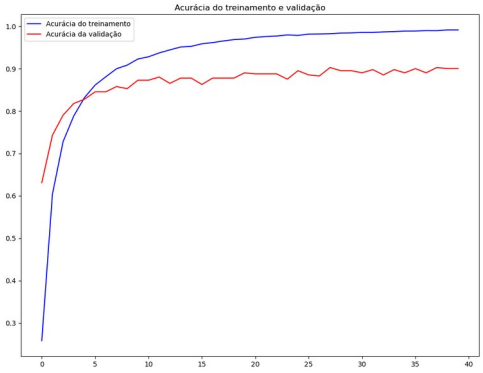
Conforme dito anteriormente foi utilizado dois tipos de data augmentation, o resultado será apresentado abaixo como maneira de comparação.

Quando utilizado data agumentation com as imagens geradas pelas fontes semelhantes a manuscritas, não houve uma melhora significativa da acurácia (0.72069) além das confusões apresentadas serem semelhantes também, e como ocorrido em todos os testes com data augmentation, ocorreu uma elevação significativa no tempo de execução dos algoritmos.

Resultado das redes com Data Augmentation

Imagens manipuladas e originais

LeNet 5	**Outra CNN**
accuracy: 0.86125	accuracy: 0.90035

Matriz de confusão	**Matriz de confusão**
[34 3 0 0 0 1 1 0 0 0 0 0] [1 26 0 0 0 0 0 2 1 0 1 1] [0 0 35 1 0 0 0 0 0 0 0 0] [0 0 0 36 0 0 1 2 0 0 0 0] [0 0 0 5 32 1 0 0 0 0 0 0] [3 0 0 0 0 24 2 0 0 0 0 0] [0 0 0 0 1 4 27 0 0 0 0 0] [0 0 0 0 0 0 0 24 0 0 2 2] [0 0 0 0 0 0 0 0 24 1 3 3] [0 0 0 0 0 0 0 0 0 30 0 0] [0 1 0 0 0 0 0 0 2 0 31 0] [0 0 0 0 0 2 0 1 4 2 2 22]	[33 3 0 0 0 1 1 0 0 0 0 0] [3 28 0 0 0 0 1 0 0 1 0 0] [0 0 35 0 1 0 0 0 0 0 0 0] [0 0 0 37 1 0 0 1 0 0 0 0] [0 0 0 3 35 0 0 0 0 0 0 0] [3 0 0 0 0 24 2 0 0 0 0 0] [0 0 0 0 0 3 29 0 0 0 0 0] [0 0 1 0 0 0 0 25 0 0 0 2] [0 0 0 0 0 0 0 0 27 2 0 2] [0 0 1 0 0 0 0 0 0 29 0 0] [0 0 0 0 0 0 0 0 0 1 32 1] [0 0 0 0 0 1 0 1 1 0 3 27]
	

Por fim, conforme solicitado no documento do moodle, foi implementado o classificador SVM com as mesmas estratégias apresentadas até o momento, e usando a rede pré treinada da Imaginet.

A acurácia apresentada pelo classificador SVM foi de 25,35% sem a estratégia de data Augmentation e 36,87% com a estratégia de data augmentation.

As matrizes de confusão e os resultados da acurácia apresentados até o momento e a seguir, nos permite verificar com clareza que a CNN que geramos com parâmetros aleatórios (ou melhor muito próximo daquilo que foi apresentado em aula) nos entrega um melhor resultado em comparação com as demais utilizadas.

SVM Sem data augmentation	SVM Com data augmentation
[8 3 21 3 1 1 1 1 0 0 0 0] [8 2 21 1 0 0 0 0 0 0 0 0] [3 0 29 2 0 0 1 0 0 1 0 0] [4 0 7 27 0 0 0 0 0 0 1 0] [2 0 16 7 10 0 0 0 1 0 2 0] [5 0 15 5 0 1 1 0 0 0 1 1] [9 0 10 8 2 0 2 1 0 0 0 0] [7 0 17 2 1 0 0 0 0 1 0 0] [5 1 13 8 0 0 0 0 1 0 3 0] [3 0 9 8 2 0 0 0 1 3 4 0] [2 1 16 3 4 0 0 0 1 0 9 0] [6 0 18 1 1 1 1 0 0 1 2 2]	[10 5 14 4 1 0 2 2 0 1 0 0] [4 6 15 1 1 0 0 2 1 0 0 2] [0 2 30 0 1 0 1 0 0 1 0 1] [1 1 7 25 0 0 0 0 2 1 1 1] [1 2 9 4 14 0 0 0 2 3 3 0] [4 0 15 2 0 3 3 1 0 0 0 1] [3 1 6 6 1 2 11 2 0 0 0 0] [3 1 16 1 1 0 0 3 0 1 0 2] [2 2 9 3 0 0 1 1 8 1 1 3] [2 1 6 5 3 0 0 0 4 5 2 2] [2 1 11 4 2 0 0 0 1 1 11 1] [4 0 13 0 1 2 2 0 3 1 2 5]

CONCLUSÃO

Foi possível notarmos grande semelhança entre os dois primeiros classificadores e o comportamento também ao utilizarmos o data augmentation trás uma significativa melhora para ambos os classificadores, também foi possível ter uma visão geral do funcionamento de cada um para conhecimento e aplicação.

O SVM implementado teve grande desvantagem sobre as redes neurais utilizadas anteriormente, com os erros concentrados nos dois primeiros meses do ano conforme fica evidente nas matrizes de confusão.

Com a analise detalhada das matrizes de confusão foi possível ainda ter uma visão de onde acontecem os erros, que também são semelhantes conforme comentado anteriormente,ao utilizar o data augmentation é necessário cautela pois o custo pode ser aumentado consideravelmente, essa ferramenta nos permitiu visualizar como se comportaria o custo dos aplicativos.