

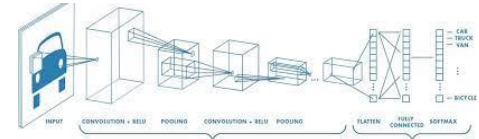
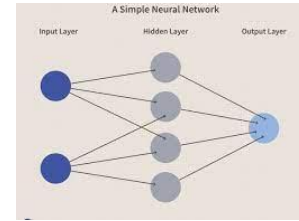
Neural Networks: When are they Useful?

An Analysis of Three Different Datasets


Michael Zhou (mgz27), Qiaochu Xiong (qx27), Brian Ling (bjl95)

Introduction

- Build a suite of neural network algorithms:
 - Fully-Connected Neural Network
 - Convolutional Neural Network
- Datasets Analyzed:
 - Music Notes Datasets (28x28 and 64x64 images)
 - Email Spam Classification Dataset
 - UCI Wine Quality Datasets (Red and White wine examples)
- Goal: Figure out which type of neural network works and does not work for each dataset.

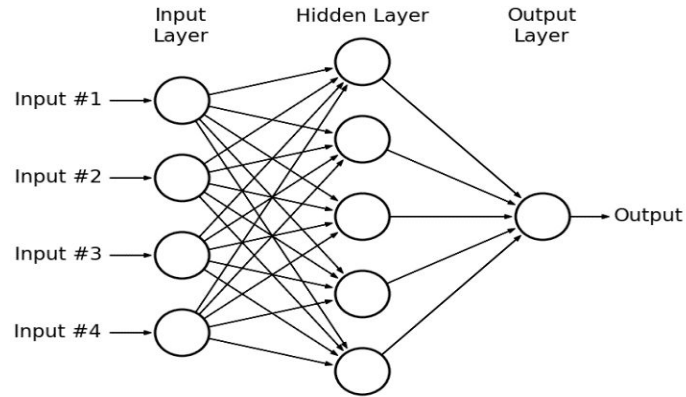


Dataset 1: Music Notes Dataset

- 2 balanced datasets of 5000 grayscale images
 - Small (28 x 28)
 - Large (64 x 64)
- Types of notes:
 - Whole
 - Half
 - Quarter
 - Eighth
 - Sixteenth
- Objective: Classify the note  used on each image
- No standardization needed since all features are uniform (grayscale pixel values)

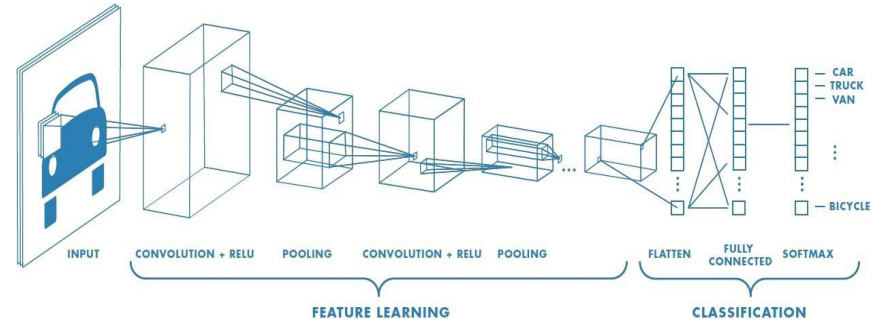


Music Notes Dataset: Models Used



Fully-Connected
Neural Network
(FCNN)

VS



Convolutional
Neural Network
(CNN)

FCNN - Experimentation Method

- MLPClassifier from Scikit-Learn
- 80-20 train-test splits
- Parameter Grid:
 - Early stopping: True, False
 - Hidden Layer Sizes: 100, 200, ..., 500
 - Activation: Relu, Tanh, Logistic
 - Learning rate: Constant, Inverse-scaling, Adaptive
 - Learning rate init: 0.0001, 0.001, 0.01, 1
- Maximum 50K iterations
- Adam optimizer (default)
- 5-fold CV
- Ran grid search and random search (20 iterations)

CNN - General Experimentation Method

- Models built using Keras
- 80-20 train-test splits
- Converted note categories (whole, half, quarter, eighth, sixteenth) into integers 0-4
- Separate CNN architectures for 28x28 and 64x64 images (next two slides) tuned manually (no grid or random search CV)

CNN Architecture (28 x 28 images)

- Conv2D - 32 filters, 3x3 filters, tanh activation, input shape (28,28,1)
- MaxPooling2D - 2x2 pool size
- Conv2D - 40 filters, 1x1 filters, tanh activation, input shape (28,28,1)
- MaxPooling2D - 2x2 pool size
- Conv2D - 50 filters, 11x11 filters, tanh activation
- MaxPooling2D - 2x2 pool size
- Flatten
- Dense layer with 5 class outputs
- Softmax activation
- Categorical cross-entropy loss + Adadelta optimizer

Model: "sequential_22"

Layer (type)	Output Shape	Param #
conv2d_45 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_39 (MaxPooling)	(None, 13, 13, 32)	0
conv2d_46 (Conv2D)	(None, 3, 3, 40)	154920
max_pooling2d_40 (MaxPooling)	(None, 1, 1, 40)	0
flatten_13 (Flatten)	(None, 40)	0
dense_13 (Dense)	(None, 5)	205
activation_13 (Activation)	(None, 5)	0

Total params: 155,445
Trainable params: 155,445
Non-trainable params: 0

CNN Architecture (64 x 64 images)

- Conv2D - 32 filters, 3x3 filters, tanh activation, input shape (64,64,1)
- MaxPooling2D - 2x2 pool size
- Conv2D - 40 filters, 11x11 filters, tanh activation
- MaxPooling2D - 2x2 pool size
- Flatten
- Dense layer with 5 class outputs
- Softmax activation
- Categorical cross-entropy loss + Adadelta optimizer

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	320
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 21, 21, 40)	154920
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 40)	0
flatten (Flatten)	(None, 4000)	0
dense (Dense)	(None, 5)	20005
activation (Activation)	(None, 5)	0
Total params: 175,245		
Trainable params: 175,245		
Non-trainable params: 0		

Music Notes Dataset: Model Comparison + Results

- CNN model far superior than FCNN in terms of test accuracy
- CNN has much slower training time
- Larger image datasets also slow down CV search and training (especially for CNNs)
- Key Takeaway: CNNs work best for image classification

Small Dataset (28 x 28 Images):

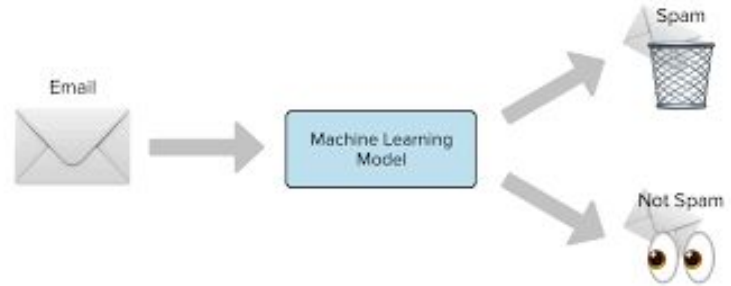
Algorithm	CV Search Method	Test Accuracy	Cross-validation Search Time (seconds)	Training Time (seconds)
Fully-Connected NN	Grid	0.88	6134.6064739227295	37.561065912246704
Fully-Connected NN	Randomized (20 iterations)	0.844	539.9790909290314	31.42546510696411
CNN	N/A (Manually tuned)	0.9340000152587891	N/A (Manually tuned)	1772.1896510124207

Large Dataset (64 x 64 Images):

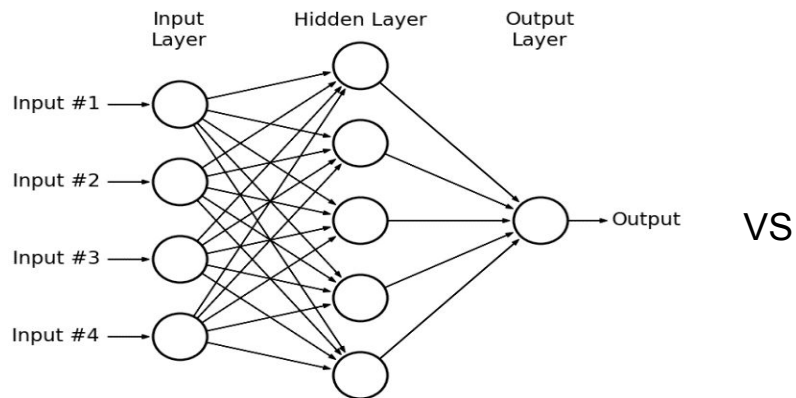
Algorithm	CV Search Method	Test Accuracy	Cross-validation Search Time (seconds)	Training Time (seconds)
Fully-Connected NN	Grid	0.811	25156.27631998062	78.14718914031982
Fully-Connected NN	Randomized (20 iterations)	0.826	2366.9517533779144	77.53696775436401
CNN	N/A (Manually tuned)	0.9599999785423279	N/A (Manually tuned)	15400.268100738525

Dataset 2: Email Spam Classification Dataset

- A dataset with 5172 emails
 - Each row represents an email
 - First column contains email names
 - 3000 columns represents 3000 words
 - Last column contains classification
- 2 classes in the last column
 - 1 represents spam
 - 0 represents not spam
- Objective: determine whether a given email is spam or not

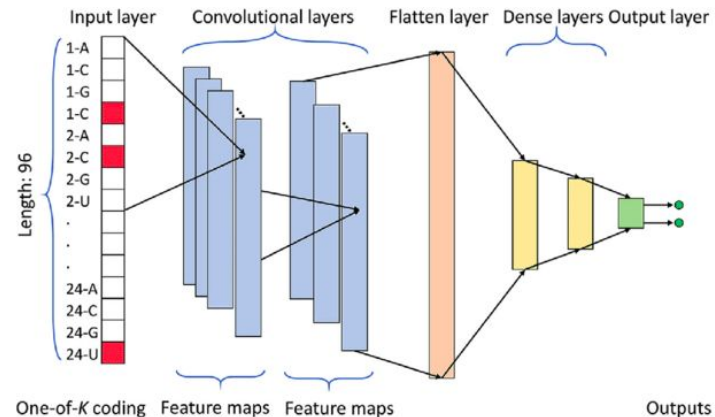


Email Spam Classification Dataset: Models Used



Fully-Connected Neural Network
(FCNN)

VS



One-dimensional Convolutional Neural
Network

FCNN - Experimentation Method

- MLPClassifier from Scikit-Learn
- 80-20 train-test splits
- Parameter Grid:
 - Early stopping: True, False
 - Hidden Layer Sizes: 100, 150
 - Activation: Relu, Tanh
 - Learning rate: Constant, Adaptive
 - Learning rate init: 0.01, 1
- Maximum 50K iterations
- Adam optimizer (default)
- 5-fold CV
- Ran grid search and random search (32 iterations)

CNN - Experimentation method

- 80-20 train-test splits
- Architecture:
 - Conv2D - 32 filters, 3x3 filters, relu activation, input shape (3000,1)
 - MaxPooling2D - 2x2 pool size
 - Conv2D - 40 filters, 11x11 filters, relu activation, input shape (28,28,1)
 - MaxPooling2D - 2x2 pool size
 - Flatten
 - Dense layer with 2 class outputs
 - Sigmoid activation
 - Binary cross-entropy loss + RMSProp optimizer
- Tuned Manually

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 2998, 32)	128
max_pooling1d (MaxPooling1D)	(None, 1499, 32)	0
conv1d_1 (Conv1D)	(None, 1489, 40)	14120
max_pooling1d_1 (MaxPooling1D)	(None, 744, 40)	0
flatten (Flatten)	(None, 29760)	0
dense (Dense)	(None, 2)	59522
activation (Activation)	(None, 2)	0

=====
Total params: 73,770
Trainable params: 73,770
Non-trainable params: 0
=====

Email Spam Classification Dataset: Model Comparison + Results

- Testing accuracies of FCNN and CNN are similar
- CNNs were trained slower compared to FCNN, but much faster comparing to Musical Notes Datasets
- FCNN selected by randomized search has the best testing accuracy

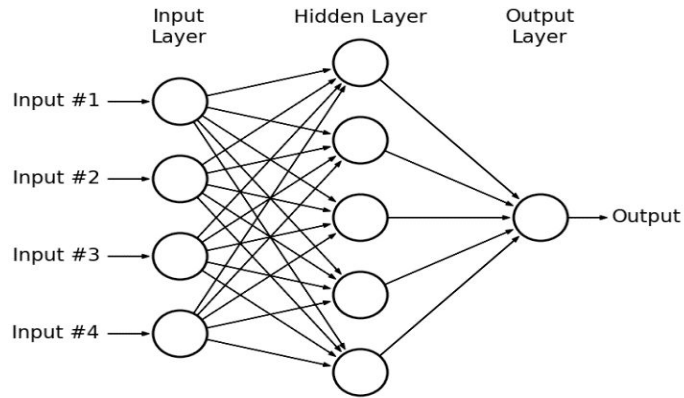
Algorithm	CV Search Method	Test Accuracy	Cross-validation Search Time (seconds)	Training Time (seconds)
Fully-Connected NN	Grid	0.9729468599033816	1817.4925224781036	10.122442722320557
Fully-Connected NN	Randomized	0.9797101449275363	2056.2531270980835	47.68662929534912
CNN	N/A (Manually tuned)	0.9787439703941345	N/A (Manually tuned)	202.64555978775024

Dataset 3: UCI Wine Quality Dataset

- 2 imbalanced datasets
 - 1599 Red Wine samples
 - 4898 White Wine samples
- 11 predictors:
 - Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol
- Goal: Predict wine quality (integer ranging from 0 to 10)

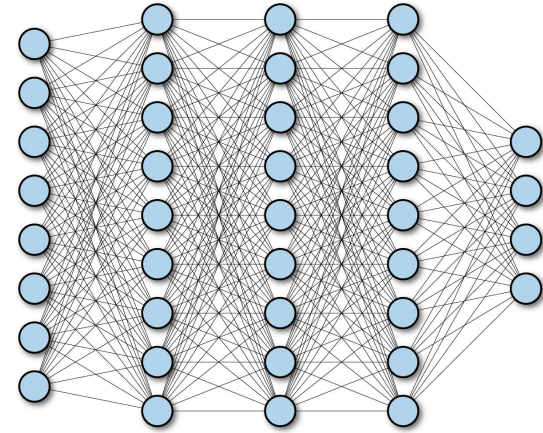


UCI Wine Quality Dataset: Models Used



Fully-Connected Neural
Network
(Single Hidden Layer)

VS



Fully-Connected Neural
Network
(Multiple Hidden Layers)

Single-Layer FCNN - Experimentation Method

- Scaled dataset using a StandardScaler
- Exact same procedure and parameter grids as FCNN model for Music Notes Dataset
- Used grid and random search

Multi-Layer FCNN - Experimentation Method

- Scaled dataset using a StandardScaler
- Used Keras tuners for random search (60 trials, 1 execution per trial)
- Random search architecture grid:
 - Dense layer with 50, 100, ..., 400 units + dropout rate 0.2
 - 1 to 12 dense layers with 50, 100, ..., 400 units + dropout rate 0, 0.1, ..., 0.5
 - Another dense 50, 100, ..., 400 units + dropout rate 0, 0.1, ..., 0.5
 - Output dense layer + softmax activation
 - Adam optimization, MSE loss
- Red wines have quality labels 3-8, white wines 3-9
- 100 maximum epochs per trial + batch size 32
- Early stopping criteria - stop if 10 iterations without validation loss improvement
- Saved best models into h5 files
- Did not produce compelling results (probably due to some bug - probably the loss function?)

UCI Wine Quality Dataset: Model Comparison + Results

- Single Hidden Layer FCNN has much higher test accuracy
- Multiple Hidden Layer FCNN underfits due to some bug
- Grid Search CV for Single Hidden Layer boosted test performance compared to Random Search CV
- Red wine models consistently outperform white wines
- White wines take longer to train
- Low accuracies overall - why?

Red Wines Dataset:

Algorithm	CV Search Method	Test Accuracy	Cross-validation Search Time (seconds)	Training Time (seconds)
Single Hidden Layer Fully-Connected NN	Grid	0.653125	2361.737888097763	3.13327693939209
Single Hidden Layer Fully-Connected NN	Randomized (20 iterations)	0.6375	151.85131907463074	15.565123796463013
Multiple Hidden Layer Fully-Connected NN	Randomized (60 iterations)	0.525	376.85603404045105	N/A

White Wines Dataset:

Algorithm	CV Search Method	Test Accuracy	Cross-validation Search Time (seconds)	Training Time (seconds)
Single Hidden Layer Fully-Connected NN	Grid	0.6479591836734694	17003.405514001846	46.954169034957886
Single Hidden Layer Fully-Connected NN	Randomized (20 iterations)	0.6183673469387755	688.1731026172638	46.393786907196045
Multiple Hidden Layer Fully-Connected NN	Randomized (60 iterations)	0.25204081632653064	804.2011730670929	N/A

UCI Wine Quality Dataset: Why Low Test Accuracy?

- Weak correlations between predictors (left)
- Imbalanced Class Distributions (right)

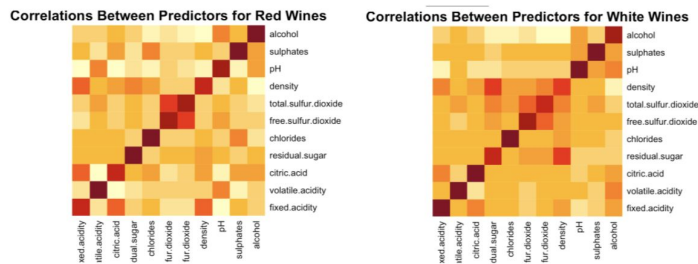


Figure 3.2: Heat maps showing correlations between predictors for red wines (left) and white wines (right). Lighter color denotes weaker relationship between two predictors.

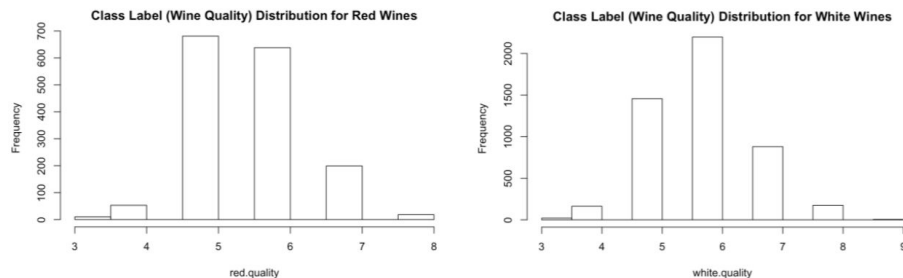


Figure 3.1: Class label distributions for red wines (left) and white wines (right)

Conclusion

- Analyzed 3 Datasets: Music Notes Classification, Spam Classification, and Wine Quality.
- CNNs work well for image-based datasets (Music Notes Classification)
- FCNN is suitable for datasets that are densely populated by zero
- Larger size -> longer training/fitting times
- Weak inter-predictor correlations + Unbalanced dataset -> Poor Generalization Performance

