Artificial Neural Networks

EEL4930 Special Topics in CISE: Applied Machine Learning – Project 3

Shane Ferrell  
Department of Computer and Information Sciences and Engineering  
University of Florida Wertheim College of EngineeringGainesville, Florida, USA  
shaneferrell@ufl.edu

Chart, bar chart

Description automatically generated*Abstract*—This paper discusses approaches to the design process of artificial neural networks for machine learning techniques. Various model configurations and learning approaches for both classification and regression tasks are explored. (*Abstract*)

Keywords—machine learning, neural networks, transfer learning (key words)

# Tools

Machine learning has been a surging field in the past decade. Machine learning practice and knowledge have become much more complex and extensive due to the increasing power of machine learning technology. In this paper, we explore the use of programming language python paired with the use of the free software libraries Scikit-Learn, NumPy, TensorFlow with Keras, and Matplotlib.

# Classification Task

## Flower Species Dataset

### Dataset Information

The data used to explore these techniques and approaches is the flower species dataset. This dataset includes 2073 photos of 10 various flower species, each of which is 300x300 pixels in 3 channels for RGB.

|  |  |
| --- | --- |
| **Flower Species** | **Label** |
| Roses | 0 |
| Magnolias | 1 |
| Lilies | 2 |
| Sunflowers | 3 |
| Orchids | 4 |
| Marigold | 5 |
| Hibiscus | 6 |
| Firebush | 7 |
| Pentas | 8 |
| Bougainvillea | 9 |

1. Class labels for the flower species dataset

The subject of each sample is generally in the center of the image. All labels are assumed to be correct and accurate.

### Data Preprocessing

To prevent overfitting of our models, we split the data into training, validation, and test sets for the supervised learning aspects of our classification task. The dataset, however, has unequal class sizes (Figure 2). The dataset must be partitioned to stratify equal proportions of classes throughout the training, validation, and test sets. The data is labeled, sufficient in size, and scaled, so it can be immediately used by machine learning algorithms and tools.

1. Class frequency in the flower species dataset

Dimensionality reduction is utilized to downsample the dataset. Some machine learning techniques can be computationally expensive if dimensionality is too high. In the case of artificial neural networks, the number of parameters grows very quickly with higher-dimensional data. The dataset in this analysis was downsampled from images of size 300x300 to 150x150 pixels, with each downsampled photo having every other column and row completely discarded.

## Neural Network

Artificial neural networks are a form of artificial intelligence loosely modeled after the system of neurons that form the brain. Neural networks are extremely complex and versatile, used for applications in many various fields of study.

The classification task at hand is suitable for a simple neural network. Even the simplest of artificial neural networks requires many design choices and hyperparameter tuning. To examine the effects of various neural network structures and hyperparameters, we will compare a range of choices to a baseline model.

The baseline model for this comparison is a multi-layer neural network that utilizes transfer learning by the application of weights from a pre-trained model. The design choices for the baseline model are listed in Figure 3.

|  |  |
| --- | --- |
| **Design Choice** | **Baseline Value** |
| pre-trained model | Xception |
| number of hidden layers | 1 |
| number of units in hidden layer | 100 |
| hidden layer activation function | SeLU |
| kernel initializer | LeCunn Normal |
| dropout | 0.2 |
| output layer units | 10 |
| output layer activation function | softmax |
| optimizer | Adam (learning rate = 0.001) |
| regularization | early stopping (patience = 10) |
| loss | sparse categorical cross entropy |
| learning configuration | mini batch |
| batch size | 32 |
| epochs | 10 |

1. Baseline model design choices and hyperparameters

Chart, line chart

Description automatically generated

1. Baseline model learning curve

The baseline model with these design choices yields an 84% accuracy in the validation set and 98% accuracy in the training set. The baseline model is overfitting. All conclusions on design choices and hyperparameter tuning will come from comparison with this baseline model’s results. The learning curve for the baseline model is displayed in Figure 4.

## Transfer Learning

Transfer learning is the transfer of knowledge from one machine learning task applied to a similar but different task. The weights from the original model are applied to the ANN with lower-level layers’ weights being frozen to keep simple information and feature detectors that can be applied to the dataset of the new task. Various pre-trained models produce different results when their knowledge is applied to the network.

## Hidden Layers

Hidden layers in an ANN are any layer of units between the input and output layers. Hidden layers perform nonlinear transformations of the inputs of the network. More hidden layers result in greater model complexity. Each hidden layer can correspond to a task or subproblem. The flowers dataset classification problem, however, is relatively simple and there is no need for a deep hidden-layer architecture as only one hidden layer is needed for most tasks.

Each hidden layer can have an independent number of units, activation function, and kernel initializer. These design choices will be kept the same for multiple hidden layers for ease of testing but will be experimented with in the case of a single hidden layer.

## Epochs & Batch Size

Batch size is the number of samples processed by a network before any weights are updated. The number of epochs is the number of times that the full dataset is processed by a network. Tuning for number of epochs and batch size is important because it affects the speed to which the network approaches the global optima.

## Checkpoints & Early Stopping

Checkpoints can prevent overfitting by choosing the update to the weights that performs best in the validation set. Early stopping is a form of checkpoint that stops training if the validation loss is increasing. The baseline model has technically no early stopping as the patience is equal to the number of epochs.

## Learning Rate

The learning rate determines the step size in moving towards the minima of the loss function. Adjusting learning rate is important in order to converge to the global optima.

# Experiment summary

Figure 5 displays the training and validation performance measures for the baseline model with each design choice changed independently.

|  |  |  |
| --- | --- | --- |
| **Transfer Learning Model** | **Training Accuracy** | **Validation Accuracy** |
| Xception | 0.9759 | 0.8404 |
| VGG16 | 1.0000 | 0.8675 |
| VGG19 | 0.9985 | 0.8223 |
|  |  |  |
| **Number of Units in 1st Hidden Layer** | **Training Accuracy** | **Validation Accuracy** |
| 1600 | 0.8680 | 0.6867 |
| 100 | 0.9759 | 0.8404 |
| 20 | 0.9608 | 0.8645 |
|  |  |  |
| **Hidden Layer Activation Function** | **Training Accuracy** | **Validation Accuracy** |
| SeLU | 0.9759 | 0.8404 |
| ReLU | 0.9397 | 0.7651 |
| tanh | 0.9759 | 0.8675 |

|  |  |  |
| --- | --- | --- |
| **Hidden Layer Kernel Initializer** | **Training Accuracy** | **Validation Accuracy** |
| Lecunn | 0.9759 | 0.8404 |
| He | 0.9774 | 0.9066 |
| Glorot | 0.9774 | 0.8524 |

|  |  |  |
| --- | --- | --- |
| **Dropout** | **Training Accuracy** | **Validation Accuracy** |
| 0.2 | 0.9759 | 0.8404 |
| 0.5 | 0.9329 | 0.5181 |
| 0.7 | 0.8658 | 0.6898 |

|  |  |  |
| --- | --- | --- |
| **Learning Rate** | **Training Accuracy** | **Validation Accuracy** |
| 0.001 | 0.9759 | 0.8404 |
| 0.003 | 0.9329 | 0.6596 |
| 0.01 | 0.5415 | 0.2861 |

|  |  |  |
| --- | --- | --- |
| **Number of Epochs** | **Training Accuracy** | **Validation Accuracy** |
| 10 | 0.9759 | 0.8404 |
| 20 | 0.9661 | 0.7590 |
| 30 | 0.9563 | 0.4217 |

|  |  |  |
| --- | --- | --- |
| **Batch Size** | **Training Accuracy** | **Validation Accuracy** |
| 32 | 0.9759 | 0.8404 |
| 64 | 0.9736 | 0.7229 |
| 10 | 0.9751 | 0.7199 |

|  |  |  |
| --- | --- | --- |
| **Presence of Checkpoints** | **Training Accuracy** | **Validation Accuracy** |
| None | 0.9759 | 0.8404 |
| Save Best Only | 0.9864 | 0.8404 |
|  |  |  |
| **Number of Hidden Layers** | **Training Accuracy** | **Validation Accuracy** |
| 1 (2073-100-10) | 0.9759 | 0.8404 |
| 2 (2073-300-100-10) | 0.9005 | 0.8614 |
| 3 (2073-1000-300-100-10) | 0.2557 | 0.2620 |

1. Performance measures for independent design choices and hyperparameters

# Discussion

The guidelines for design choices and hyperparameters for building a neural network are generally applicable to our dataset. The neural networks with the VGG16 pre-trained model, 20 units in the hidden layer, activation function tanh, He initialization, and the 2-hidden-layer architecture all performed surprisingly better than the baseline network. There is no guarantee that all of these design choices and hyperparameters would have a better performance than the baseline network though, as each choice in a network can make a significant difference in the network’s structure.

Nevertheless, the test set performance of the baseline network and the network with design choices and hyperparameters that independently produced better performance than the baseline are shown in Figure 6.

|  |  |  |
| --- | --- | --- |
| **Network** | **Test Set Accuracy** | |
| Baseline | | 0.87 |
| Independently Ideal  Design & Hyperparameters | 0.11 | |

1. Test set accuracy for the baseline model and network with each indepent choice

# Regression Task

## Car Detection Dataset

### Dataset Information

The data used to explore these techniques and approaches is the car detection dataset. This dataset includes 734 photos of street traffic, each of which is 380x686 pixels in 3 channels for RGB. The subject of each sample is not guaranteed to be in the center of the image. In fact, not every sample contains a car. All labels are assumed to be correct and accurate.

### Data Preprocessing

To prevent overfitting of our models, we split the data into training, validation, and test sets for the supervised learning aspects of our classification task. The data is labeled, sufficient in size, and confined to a fixed range, so it can be immediately used by machine learning algorithms and tools.

## Neural Network

The regression task at hand is suitable for a deep learning neural network. Deep learning artificial neural networks require complex design choices and hyperparameter tuning. We can experiment with various aspects of the network’s structure and hyperparameters to find the design with the highest performance for this task and dataset.

Transfer learning is useful for this task because lower layers can be reused by freezing the weights for the input layer. These layers contain information that help detect simple features such as edges. This information is also helpful for our task because our goal is to, at a very low level, distinguish the cars from the background by finding its edges.

|  |  |
| --- | --- |
| **Design Choice** | **Baseline Value** |
| pre-trained model | Xception |
| number of hidden layers | 2 |
| number of units in hidden layers | 100 (1st), 50 (2nd) |
| hidden layers activation function | SeLU |
| kernel initializers | LeCunn Normal |
| dropout | 0.5 |
| output layer units | 10 |
| output layer activation function | ReLU |
| optimizer | Adam (learning rate = 0.0003) |
| regularization | early stopping (patience = 10) |
| loss | mean squared error |
| learning configuration | mini batch |
| batch size | 32 |
| epochs | 5 |

1. Best-performing netork design choices and hyperparameters

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

1. Learning curves for training the neural network

The network with these design choices yields an 79% accuracy in the validation set and 64% accuracy in the training set. All conclusions on design choices and hyperparameter tuning will come from experimenting with these design choices hyperparameters, not from any extensive cross-validation strategies. The learning curve for training the network is displayed in Figure 8.

## Test Set Performance

There are no target labels for the data in the test set, so we can visually examine the predicted boundaries by the network to judge its performance. Figures 9 shows the first 8 shuffled samples from the test set with the predicted boundary box.

# Discussion

It appears that the network fails to create a boundary with the car completely inside its boundaries a majority of the time. The model appears to perform best with a car near the left edge of the screen.

A picture containing text, sky, outdoor, green

Description automatically generatedA picture containing tree

Description automatically generated A picture containing text, sky, tree, athletic game

Description automatically generated A picture containing sky, tree, outdoor, athletic game

Description automatically generated A picture containing text, sky, tree, outdoor

Description automatically generated A picture containing text, sky, athletic game, outdoor

Description automatically generated A picture containing text, sky, tree, outdoor

Description automatically generated A car driving on a road

Description automatically generated with low confidence

1. Test samples with predicted boundary boxes

# Conclusions

Artificial neural networks are a suitable strategy for both classification and regression tasks. Various design choices for the structure and values for hyperparameters of an artificial neural network can have a great impact on its performance.