Machine Learning Engineer Nanodegree

Capstone Proposal

Andrew Smith
October 3rd 2017

Proposal

Domain Background

In the 1999 paper *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables (Blackard et al, Computers and Electronics in Agriculture 24, 1999)*, Blackard and Dean used an ANN with a single hidden layer of 120 units to predict forest cover type for data drawn from four wilderness areas with a predictive accuracy of 70.52%. The dataset used has 581,012 instances, each with 10 numerical values and 44 categorical values – 4 wilderness types and 40 soil types – and each is labeled with one of 7 forest cover types such as Spruce/Fir or Lodgepole Pine. Since the traditional methods of recording forest cover can be expensive a method of reliably predicting cover type from other variables could be very useful.

Since that paper was written a number of advances have been made in both hardware – with the wide availability of GPUs suitable for use in ANNs – and technique – such as the use of RLUs, Dropout, Glorot Initializers and the wide use of Softmax Activation on the output layer – which have allowed the construction of ANNs with greatly improved accuracy. While these results are more well known in areas such as computer vision and natural language processing it is interesting to examine their impact on a simpler classification problem

Problem Statement

As stated in the original paper *"Accurate natural resource inventory information is vital to any private, state, or federal land management agency. Forest cover type is one of the most basic characteristics recorded in such inventories. Generally, cover type data is either directly recorded by field personnel or estimated from remotely sensed data. Both of these techniques may be prohibitively time consuming and/or costly in some situations."* (Blackard et al 1999)

The purpose of this project is to assess the improvement in accuracy in multi-class classification between a network based on the one in the original paper and a network constructed using techniques impractical, unavailable or simply not in common practice at

the time. It will do so by constructing a reference network based on the one in the 1999 paper and a series of more complex networks based on the techniques mentioned above, using the same set of numerical and categorical data used in the original study.

Datasets and Inputs

The dataset used for this project will be the Forest Cover Type dataset available from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/datasets/Covertype. It consists of 581,012 instances of data each with 54 attributes and one of 7 classifications. There is no missing data and the categorical items in the attributes are already one hot encoded. The data consists of 10 numerical features and 44 One Hot Encoded categorical features (4 types of wilderness area and 40 soil types). The data is quite heavily skewed towards two of the categories representing 83.22% of the data between them. The breakdown is as follows:

The data has 581012 rows
There are 211840 instances of Spruce/Fir representing 36.46% of the total
There are 283301 instances of Lodgepole Pine representing 48.76% of the total
There are 35754 instances of Ponderosa Pine representing 6.15% of the total
There are 2747 instances of Cottonwood/Willow representing 0.47% of the total
There are 9493 instances of Aspen representing 1.63% of the total
There are 17367 instances of Douglas-fir representing 2.99% of the total
There are 20510 instances of Krummholz representing 3.53% of the total

In the original study the training data was selected to ensure that equal numbers of examples in each category were used for training and validation. This severely limited the amount of data available for training. In this project the data will instead be stratified using scikit-learn's StratifiedShuffleSplit via the stratify parameter of the train_test_split function to ensure that each category is represented in the test and training data in proportions equal to their proportion in the overall data. This will maximise the data available for training. If the stratification of the data leads to significant underperformance on the underrepresented categories other methods of dealing with skewed categorical representations such as random undersampling or random oversampling will be considered.

The numerical data in the dataset will be scaled to values between 0 and 1 before training and evaluation.

Solution Statement

Firstly the dataset will be pre-processed and split as detailed above. As in the original paper, all 54 features of the data will be used. An ANN similar to the one in the 1999 Blackard and Dean paper will be constructed and trained. Next a number of ANN models using 'modern'

techniques will be constructed. Each model will use 1 or more hidden layers. The input activation will be linear. The hidden layer activation will be ReLU and the output layer will use Softmax. A Glorot Uniform Initializer will be used to initialize the unit weights. Nesterov Momentum will be used by the Stochastic Gradient Descent Optimizer and the loss function will be Categorical Cross Entropy. Initially models using similar numbers of units per layer as seen in the original study, specifically 60, 120 or 240 will be used as well as larger models with 540, 1080, 1620 and 2160 units per layer (these representing 10, 20, 30 and 40 units per variable in the data). Each model will have 1 to 4 fully connected hidden layers. Each fully connected hidden layer will initially have a dropout rate of 0.5. Alternatives to this initial set of hyperparameters will be explored during refinement.

Each of these models will be trained against 80% of the original data. After training Its accuracy will be evaluated against the remaining 20% with the goal of being able to accurately predict the Forest Coverage type from the available features. Their accuracy will be compared to the model based on the one in the original study.

In the original study 1000 Epochs were used to train the models. To keep the computing time to a manageable level this project will initially try to use just 100 Epochs for training. if the models are underperforming with that number of epochs then the use of a larger number of epochs will be examined. Additionally, once the most effective model has been chosen then larger number of epochs may be used during the refinement phase.

To assess the impact of using stratification of the data as opposed to using equal numbers of each category in training, the performance of the resulting solutions against each category will be assessed.

Benchmark Model

A model will be constructed which is close to the model used in the original study.
It will have a single hidden layer of 120 units. Both the hidden and output layers will use sigmoid activation functions. The Stochastic Gradient Descent Optimizer will use the same learning rate and momentum as in the original study (0.05 and 0.5) and the loss function will be mean squared error. As stated above this model will initially only be trained over 100 Epochs using 80% of the available data. The accuracy of the model will be evaluated against the remaining 20%. If this model is able to reach a level of accuracy comparable to that found in the original study then it will be suitable for use as a benchmark against which the solution models can be evaluated.

Evaluation Metrics

Since accuracy is the metric used in the original study the models will be evaluated based on their accuracy against held back testing data. Additionally the accuracy of each model against the training data will be evaluated to help identify instances where the model is overfitting.
Since overall accuracy on a dataset skewed toward certain categories can hide poor accuracy on underrepresented categories, care will be taken to also assess accuracy of

each model against unseen data of *each individual* category to ensure that they are not heavily underperforming on those categories with less available data.

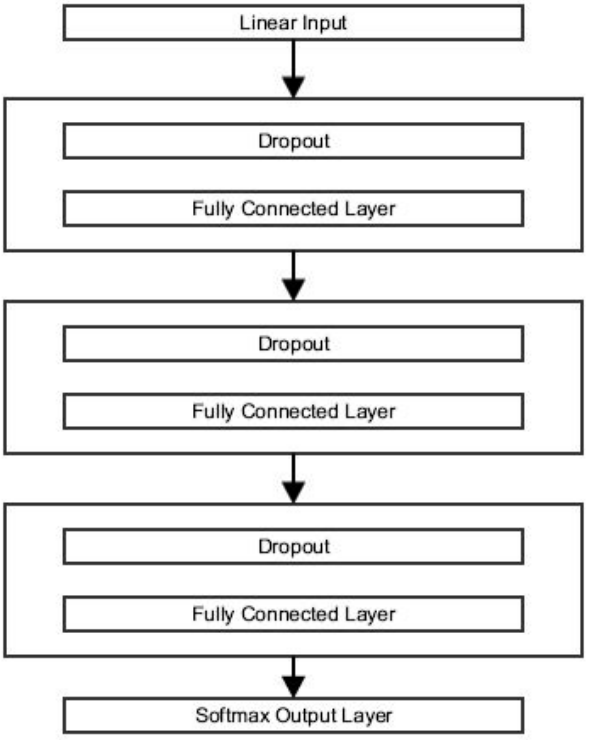
Project Design

1. The dataset will be acquired from UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/datasets/Covertype.
2. The data will be prepared as detailed in the Datasets and Inputs section. As this is a dataset from a previous study relatively little preprocessing is required.
3. A reference ANN will be constructed and trained based on the network architecture described in the original paper. The target is to reach a similar accuracy to the network from the original paper but it is not expected that the results will be identical. All comparisons will be against this network and **not** the results in the original paper. Since the data is skewed towards certain classes the accuracy of the model in each class will be assessed and some effort will be made to address any issues related to the skewed data by using techniques such as Random Undersampling, Random Oversampling or weighted samples.
4. A systematic exploration of the accuracy of models of varying complexity will be carried out against the available data. Each model will be trained and tested using the same data. Models with 60, 120, 240, 540, 1080, 1620 and 2160 units per layer, with a dropout rate of 0.5 applied to each layer and from 1 to 4 hidden layers in total will each be trained and evaluated against the available data. The models will each use the hyperparameters mentioned in the solution statement. Again since the data is skewed towards certain classes the accuracy of the model in each class will be assessed and some effort will be made to address any issues related to the skewed data by using techniques such as Random Undersampling, Random Oversampling or weighted samples. The accuracy of each model against both the training and testing data will be tabulated and presented graphically so that the most effective model can be selected.
5. The most accurate model will be selected and an attempt will be made to tune the hyperparameters to get the most accurate possible final model.
6. The results of the training runs will be presented and analysed

The models will be constructed using Keras with a Tensorflow back end. The code will be written to run on the FloydHub cloud machine learning service. This will allow the models to be evaluated using high end GPUs and will allow the simultaneous evaluation of several models. Each model will produce an output report detailing it's evaluated accuracy and loss against both the training and testing data.

To enable the code to be evaluated locally without modification, the configuration for a docker image and the code and instructions for running it will be provided.

Example of a Solution ANN Architecture



Original Paper ANN Architecture