Project Overview Athletic competitions are an interesting space to explore state-of-the-art learning methods because a large amount of relevant data is continuously produced and strategic decisions that account for this new data are often time-sensitive. In this project, we will explore how new learning methods can be used to improve training time and accuracy for models that predict how many yards will be gained on an NFL running play given an instantaneous snapshot of the state of the game. The data set containing the game state is provided for a Kaggle competition in collaboration with the NFL [4]. Game state includes information such as the position and orientation of players on the field, the current weather, and the score of the game.

Because the true density function for the amount of yards gained on any given running play is likely to be highly non-linear, we will train a neural network to perform multinomial logistic regression. In order to do so, the final layer of the neural network will use the softmax function and the network will be trained with the cross-entropy loss function. This model will allow us to interpret the output of the neural network as a discrete probability distribution.

For the Kaggle competition, a model is evaluated based on the Continuous Ranked Probability Score of its predicted cumulative distribution function. We will use the probability mass function given by our multinomial logistic regression to compute the corresponding CDF to evaluate our system on this metric. In addition, we will evaluate the wallclock time it takes to train our model because it is important to incorporate new game data into models for adjusting strategy as a game progresses.

Techniques In order to train the model we utilize standard Stochastic Gradient Descent (SGD) as our baseline. However, since models need to be retrained frequently (after every game - or even after every possession!) to match player and strategy development it is desirable to reduce training time as much as possible, while not sacrificing accuracy. Large data sets not only impose a significant memory constraint on processors, but require trading off accuracy (large batches) vs speed and robustness (small batches). To improve training we investigate layering two approaches:

1) Local convergence improvement: We aim to measure the effectiveness of variance reduction techniques, i.e. SVRG [1] in order to reduce the necessary training time until convergence.
2) Distributed learning: By spreading learning across several machines (or locally across cores) we hope to distribute computational load and storage costs.
   We explore two sub-methods:
   a) Parallel SGD [2]: This method parallelizes the Gradient Descent procedure by partitioning the data across processors. At each iteration they compute the local gradient and then synchronize to average the gradients in order to arrive at the same globally shared model. This procedure is mathematically equivalent to (Batched Stochastic) Gradient Descent) but distributes load.
   b) Simulated Parallel SGD [3]: Rather than sharing the load and synchronizing in each iteration, in this method each processor trains its own model on a random partition of the data. Finally, all learnt models are averaged. This simulates the batched SGD by training only for the length of a single partition, yet learning a model based on the entire dataset.

We will evaluate the performance of these training procedures against the baseline SGD as outlined in the Experiment Plan.

Experiment Plan (Florian)

Hypothesis:

Training with SGD on distributed copies concurrently, while locally reducing variance by the use of SVRG will achieve faster convergence. At the same time, we expect averaging multiple discrete models to result in a more robust testing error. We want to ascertain whether the variance reduction improves usability. Parallelizing SGD reduces wall-clock time, but not total CPU time (slight increase due to synchronization).

Proxy:

In order to predict the yardage gain per play we rely on a collection of game statistics that are supposedly indicative of the success of a play. Accuracy is expressed in terms of a Continuous Ranked Probability Score. In order to compare the performance of the training algorithms we will measure expired wall-clock time in relation to achieved accuracy and vice versa.

Protocol:

We will implement a Neural Network using TensorFlow (structure TBD) and train it using a single process SGD implementation as a baseline. We will compare this baseline to the parallel SGD training as well as the simulated distributed SGD algorithm

Expected results:

We expect both parallel SGD models to reduce the wall-clock time necessary to achieve comparable accuracy to the baseline SGD. The parallel distributed SGDs performance might be bottlenecked by the synchronization efforts. The simulated version alleviates this issue and should therefore achieve faster convergence, but might incur lower accuracy as it only approximates the true calculation.

Experiment Plan (Matt)

Hypothesis The model will perform better on the accuracy metric (CRPS) when the training data and test data has been cleaned to remove basic inconsistencies (e.g., plays, and thus data recorded for plays, can go in two directions on the field even though the fields are completely symmetric).

Proxy We will directly use the CRPS to test this hypothesis.

Protocol We will train the two different models on two different training data sets: an unmodified data set and a data set that has been cleaned to remove basic inconsistencies. We will then measure the CRPS for each model on the test corresponding test data sets.

Expected Results We expect that the CRPS will be higher for the data set that has been cleaned to remove basic inconsistencies.

References:

[1] Johnson, Rie, and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction." *Advances in neural information processing systems*. 2013.

[2] Mcdonald, Ryan, et al. "Efficient large-scale distributed training of conditional maximum entropy models." *Advances in Neural Information Processing Systems*. 2009.

[3] Zinkevich, Martin, et al. "Parallelized stochastic gradient descent." *Advances in neural information processing systems*. 2010.

[4] https://www.kaggle.com/c/nfl-big-data-bowl-2020/overview/evaluation