GilesStrong/QCHS-2018

RECENT DEVELOPMENTS IN DEEP-LEARNING APPLIED TO OPEN HEP DATA

Giles C. Strong

giles.chatham.strong@cern.ch @giles_c_strong

I—INTRODUCTION

- The 2014 Higgs ML Kaggle challenge was designed to stimulate outside interest in HEP problems
- The data contains simulated LHC collision data for Higgs to di-tau and several background processes
- Participants were tasked with classifying the events in order to optimise the Approximate Median Significance
- The competition was highly successful:
  - 1785 teams competed
  - The winning team achieved a score of 3.806 (higher is better)
  - It introduced new methods to HEP
  - Produced more widely used tools, such as XGBoost

- Now, it forms a nice HEP specific dataset for testing new methods
- Here we use it to show the cross-domain applicability of several recent developments in deep learning
- We achieve comparable performance (3.818) with a much more lightweight solution in terms of train time and resource requirements

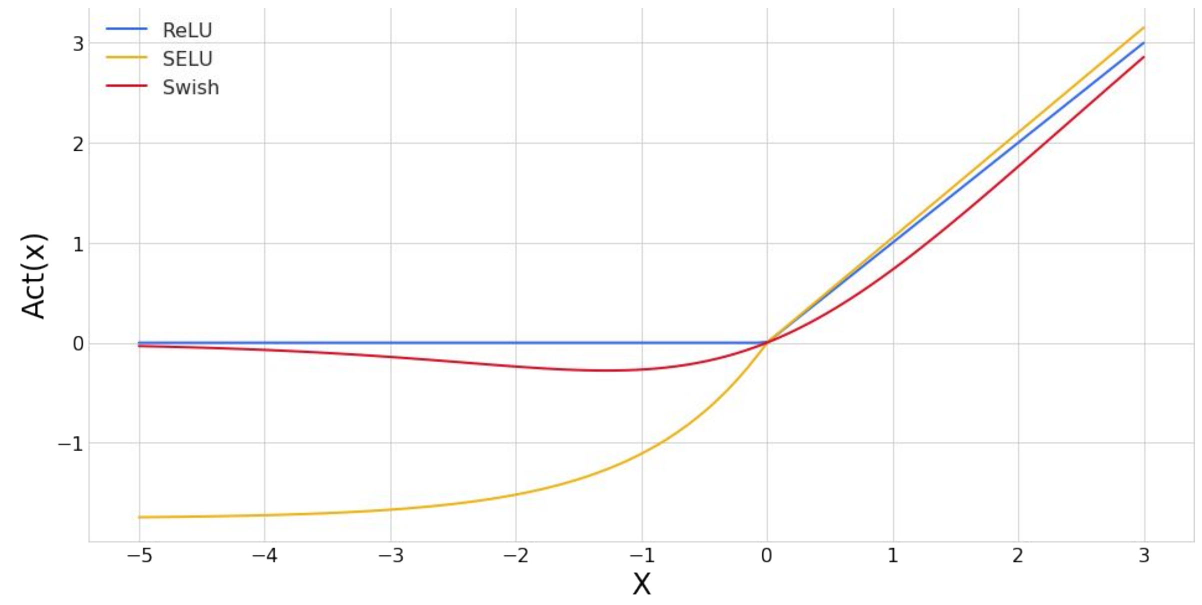
II—STARTING SOLUTION

- An ensemble of 10 Deep Neural Networks (DNNs) are trained via stratified cross-validation of 80% of the labelled data
- Each network consists of:
  - 4 hidden layers of 100 neurons with He initialisation
  - ReLU activations
  - A single sigmoid output with Glorot initialisation
- ADAM is used to minimise the binary cross-entropy of class predictions
- Batch size of 256
- The Learning Rate (LR) is optimised via a LR Range Test (Smith, 2015)

- This ensemble achieves a starting score of 3.631 (3.419 when only a single DNN is used)

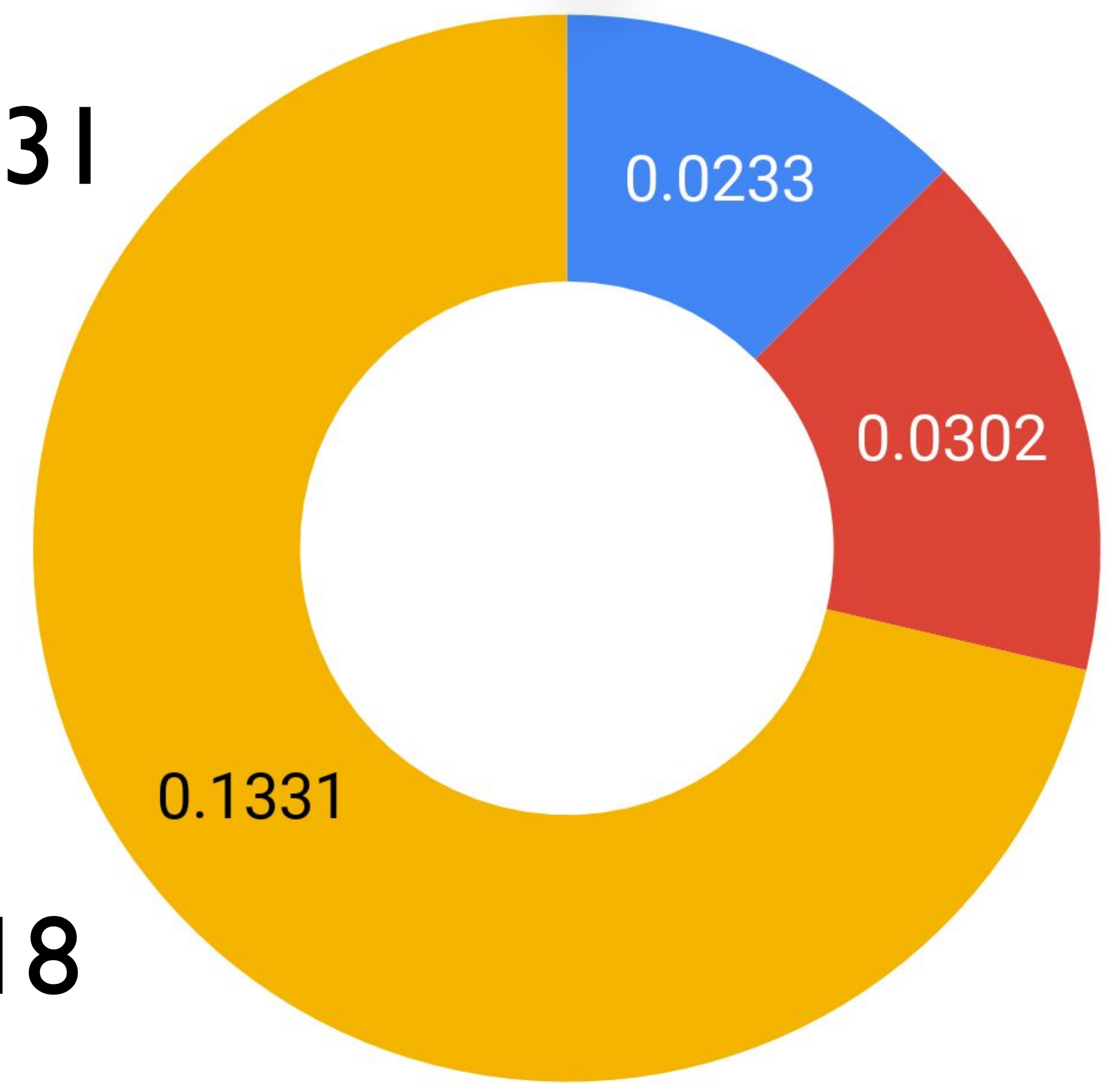
III—ACTIVATION FUNCTION

- Whilst ReLU is a common activation function, newer ones are continually being introduced
- SELU uses carefully derived scaling coefficients allow networks to self-normalise, removing any need for batch normalisation
- Swish, found via reinforcement learning, provides a region of negative gradient
- Swish was found to outperform both ReLU and SELU, improving the score to 3.654



START SCORE: 3.631

IMPROVEMENTS:



FINAL SCORE: 3.818

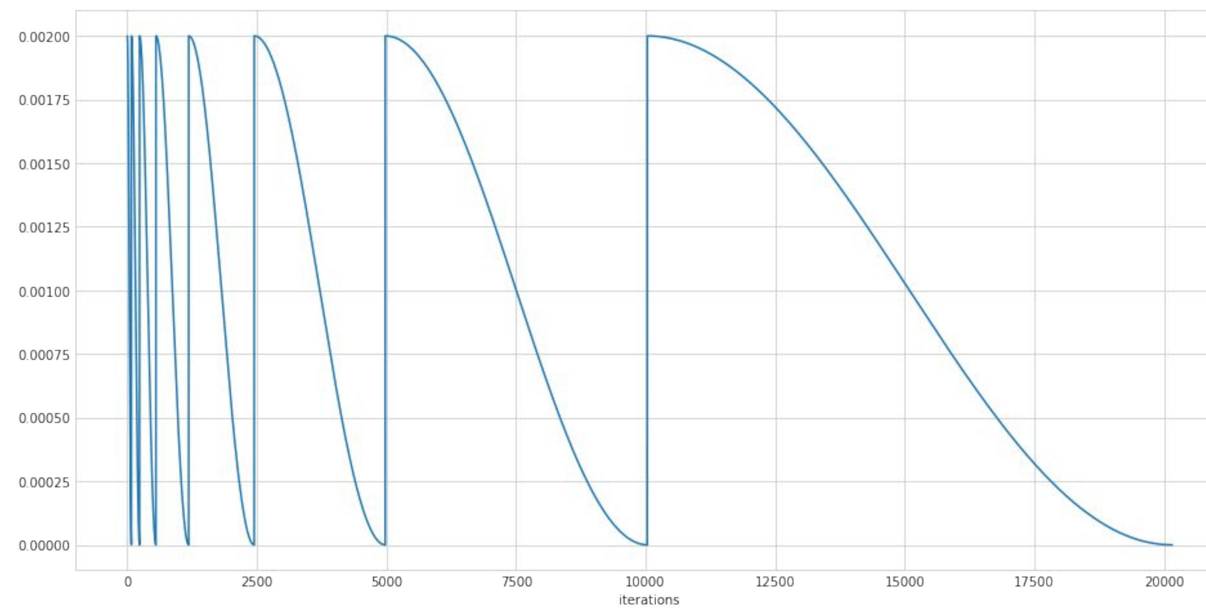
TRAIN TIME: 2 HOURS

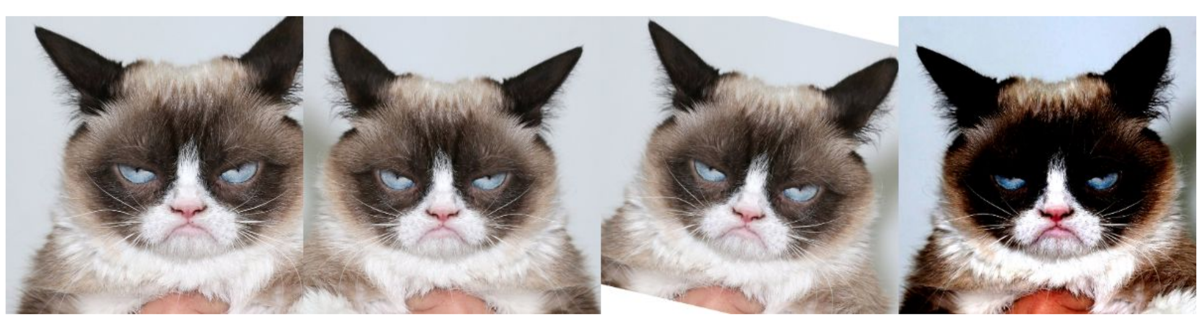
TEST TIME: 1.5 HOURS

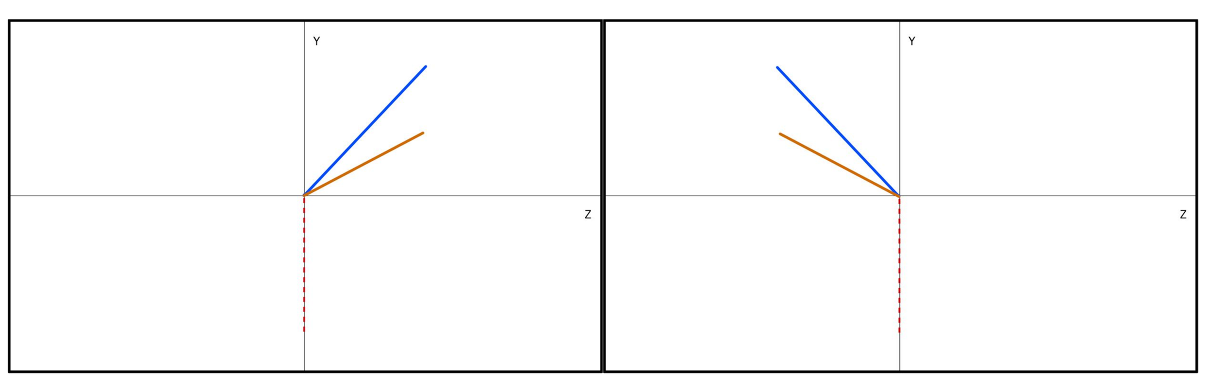
[INTEL I7-6500U - <8GB RAM]

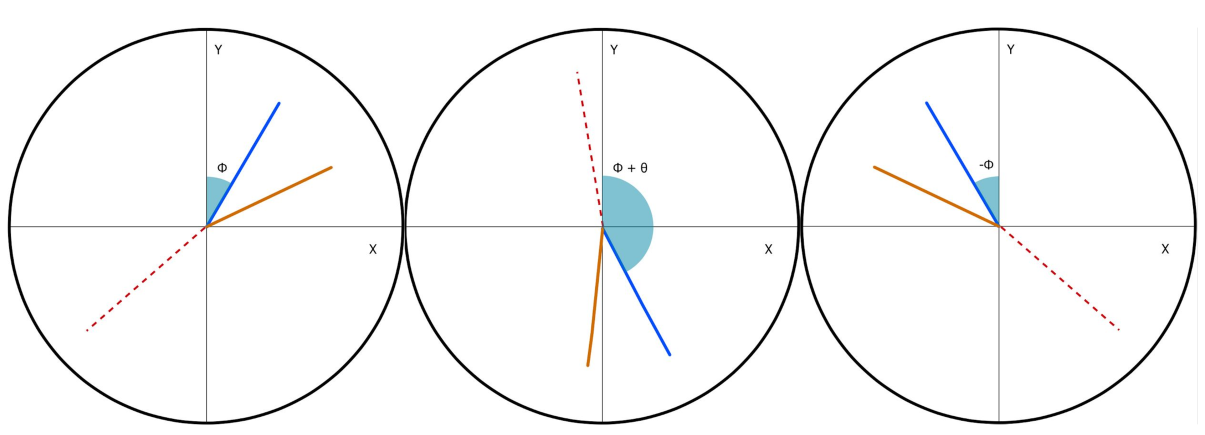
IV—LR SCHEDULE

- A common training technique is to decrease the learning rate when validation loss plateaus, to allow training to continue
- This can be automated by annealing the learning rate according to some function, such as a cosine
- Loshchilov & Hutter (2016) suggest annealing over several cosine functions, restarting once the LR drops to zero
- Huang et al. (2017) find that this method of warm restarts allows the network to explore many local minima across the loss surface.
- Implementing it here improved the score to 3.685









V—DATA AUGMENTATION

- A common technique in image classification is to apply 'class-preserving transformations' - Flips, crops, rotations, colour adjustments, et cetera
- These can be used to create augmented images to artificially increase the amount of available training data—Train-time augmentation
- The transformations can also be applied at test time by taking the average prediction over a set of augmented copied of each data point—Test-time augmentation
- Such class-preserving transformations exist in HEP; rotation of the event in azimuthal angle, and flipping it in the transverse and longitudinal axes
- Applying data augmentation during both training and testing brings the final score to 3.818

RUN IT YOURSELF

- Repository at github.com/GilesStrong/QCHS-2018
- Checkout or Fork
- Run in Binder or Docker
- Includes a more detailed presentation

BIBLIOGRAPHY

- Smith - *Cyclical Learning Rates for Training Neural Networks*, 2015, arXiv:1506.01186
- Loshchilov & Hutter - *SGDR: Stochastic Gradient Descent with Warm Restarts*, 2016, arXiv:1608.03983
- Huang *et al.* - *Snapshot Ensembles: Train 1, get M for free*, 2017, arXiv:1704.00109
- Klambauer *et al.* - *Self-Normalizing Neural Networks*, 2017, arXiv:1706.02515
- Ramachandran *et al.* - *Searching for Activation Functions*, 2017, arXiv:1710.05941