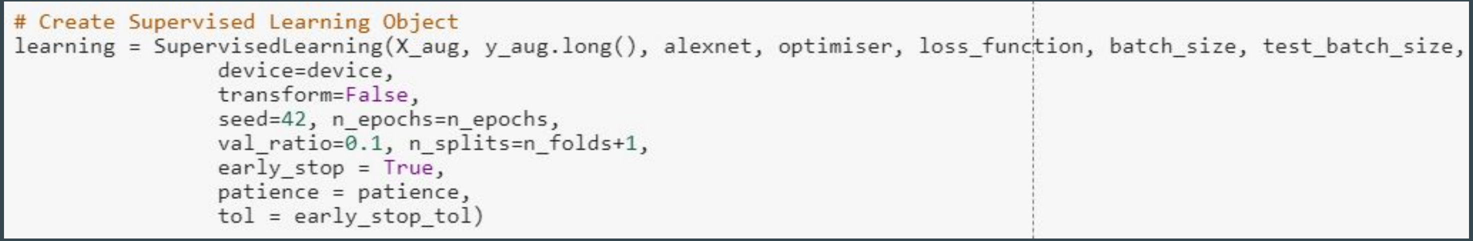
ACSE-4 Mini Project #4

Ye Liu, Oliver Boom, Mingrui Zhang, Deborah Pelacani Cruz
May 24, 2019

Project Aim

To create and implement a neural network to maximise the prediction accuracy of Kuzushiji characters by training on the KMNIST Dataset

Training Approach / Sustainability



```
# Create Supervised Learning Object
learning = SupervisedLearning(X_aug, y_aug.long(), alexnet, optimiser, loss_function, batch_size, test_batch_size,
                device=device,
                transform=False,
                seed=42, n_epochs=n_epochs,
                val_ratio=0.1, n_splits=n_folds+1,
                early_stop = True,
                patience = patience,
                tol = early_stop_tol)
```

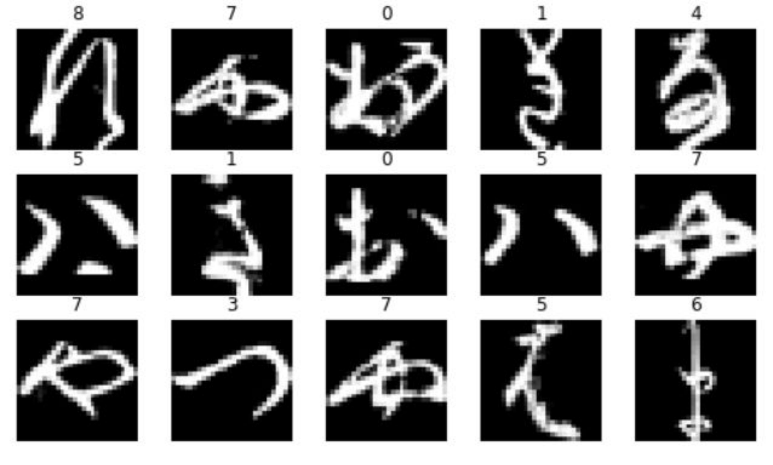
‘SupervisedLearning’ wrapper class

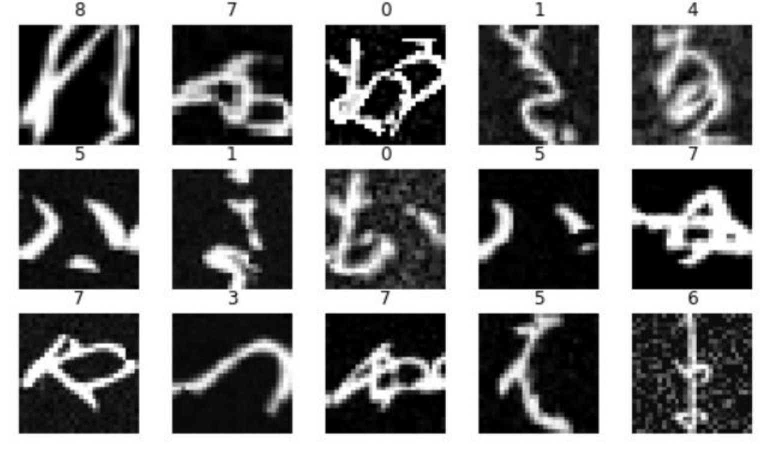
- Sustainability and Coherence
- Parameter Tuning
- Input models, optimisers and loss functions
- Early stop
- In-built Data Augmentation

- Sharing models on drive
- Github integration
- Working on different Colab Notebooks
- Saving function to save model and parameters
- Easy to perform Transfer Learning to continue training our own model

Data Pre-processing and Augmentation

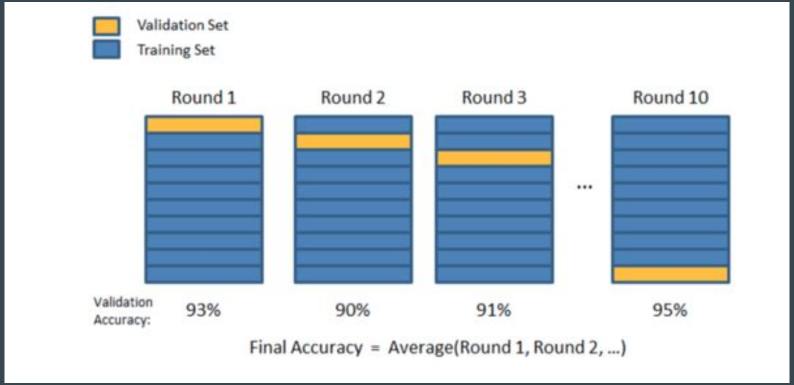
- Exploratory Data Analysis (EDA)
- Normalisation based on training set
- PyTorch transforms (class in-built)
    - Random Rotation
    - Random Crop
- Albumentation transforms (external)
    - Blur
    - Noise
    - Distortions





Validation

- Large dataset: balanced data split
- Hold-out validation (quicker!)
- K-Fold validation for better estimation of generalised error (much slower!)

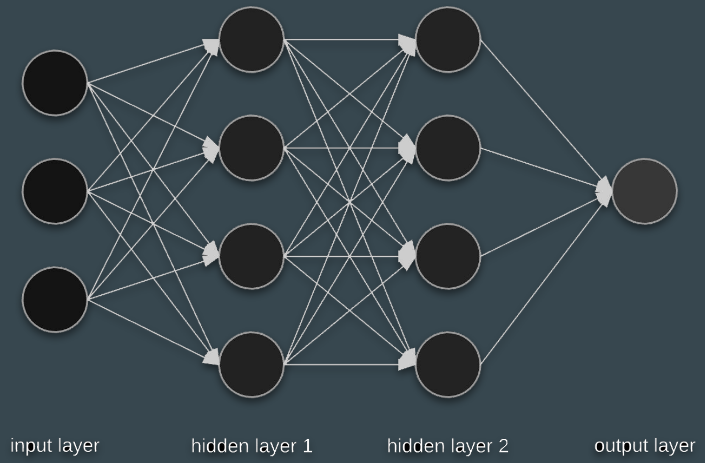


Hyperparameter Tuning on the AlexNet

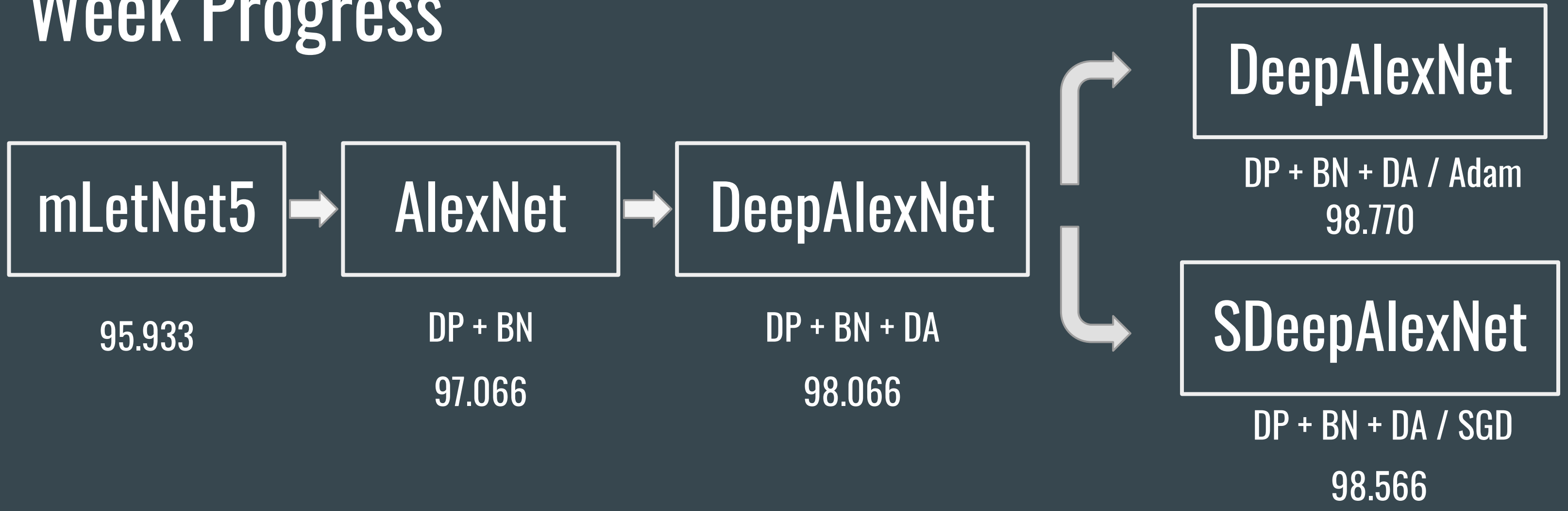
- Optimiser: Adam vs Stochastic Gradient Descent
- Learning Rate: 1e-2, 1e-4
- Adam unstable on larger learning rate
- Weight Decay: 0 , 1e-3, 1e-4, 1e-5

Neural Network Architectures

- Modifications made to AlexNet and LeNet5 to improve validation accuracy
    - AlexNetMod
    - DeepAlexNet
    - SDeepAlexNet
    - mLeNet5
- Increase in network depth
- Regularisation with dropout and batch normalisation
- Ensembles



Week Progress



mLetNet5

95.933

AlexNet

DP + BN

97.066

DeepAlexNet

DP + BN + DA

98.066

DeepAlexNet

DP + BN + DA / Adam

98.770

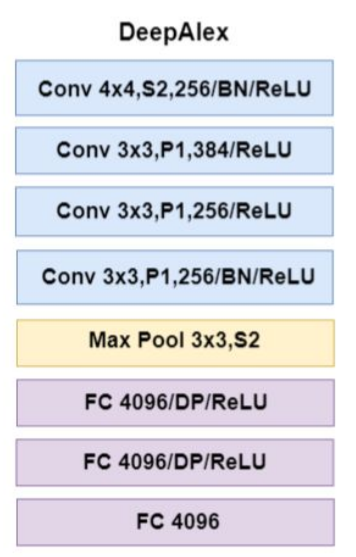
SDeepAlexNet

DP + BN + DA / SGD

98.566

Highest Scoring Models

- After validation, trained on full dataset
- Transfer Learning on previously ran models -- time saving!!
- DeepAlex and SDeepAlex with best results



DeepAlex

Conv 4x4,S2,256/BN/ReLU

Conv 3x3,P1,384/ReLU

Conv 3x3,P1,256/ReLU

Conv 3x3,P1,256/BN/ReLU

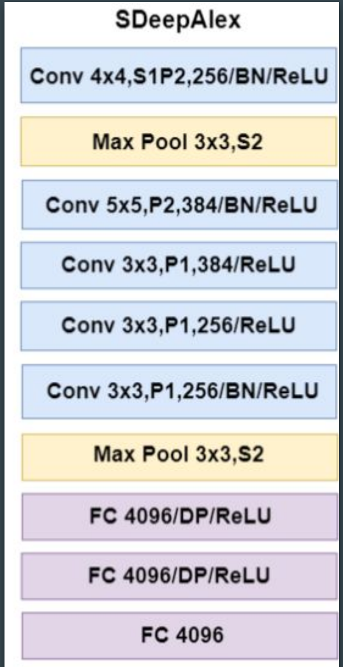
Max Pool 3x3,S2

FC 4096/DP/ReLU

FC 4096/DP/ReLU

FC 4096

(a) DeepAlexNet



SDeepAlex

Conv 4x4,S1P2,256/BN/ReLU

Max Pool 3x3,S2

Conv 5x5,P2,384/BN/ReLU

Conv 3x3,P1,384/ReLU

Conv 3x3,P1,256/ReLU

Conv 3x3,P1,256/BN/ReLU

Max Pool 3x3,S2

FC 4096/DP/ReLU

FC 4096/DP/ReLU

FC 4096

(b) SDeepAlexNet

DeepAlex	SGD 50 Epochs Transfer Learning:, last layer training with external ‘albumentation’ transforms (20 epochs)	Batch Normalisation Dropout Weight Decay 1e-4	97.90%
DeepAlex	Adam 50 Epochs In-built data augmentation		98.77%
SDeepAlex	SGD 50 Epochs In-built data augmentation Weight Decay 1e-4		98.57%

Limitations and Further Work

- Time and Computational Power!
- Explore different activation and loss functions
- Tune other hyperparameters
- Full-run with ‘albumentation’ DA instead of transfer learning
- Explore other well-established models and ensemble them