Kaught22 **for image classification**

Team Katch22: Harsh Upadhyay (hu3), Suguman Bansal (sb55)
COMP540: Rice University

Problem

- Design an image-classifier for images of size $32 \times 32 \times 3$
- 50K training images, 100K test images

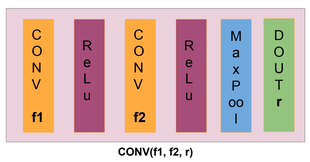
Our Approach

- Kaught22 A convolutional neural network for image classification
- Set up on **EC2**: An extremely optimized experimental set up for fast computation

Building Blocks

CONV Block(f1, f2, r)

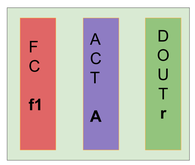
- **f1**: Filters for first CONV layer
- **f2**: Filers for second CONV layer
- **r**: Dropout ratio



CONV(f1, f2, r)

FC Block(f1, A, r)

- **f1**: Fully connected layer with $f1$ outputs
- **A**: Activation function
- **r**: Dropout ratio



FC(f1, A, r)

- Kernel size 3×3 for CONV Layer
- Stride size 2×2 for MaxPool Layer
- Nesterov's Gradient Descent (NGD) in CONV and FC layers

Kaught22 Architecture

$$\mathbf{CONV(32, 64, 0.2) - CONV(64, 64, 0.2)-}$$

$$\mathbf{-FC(512, ReLu, 0.5) - FC(10, SoftMax, 0)}$$

Implemenation details

Data Preprocessing

- Normalize values
- Shift images with mean of training set
- Shift images by 10% in vert. and hori. direction
- Flip images along vert. and hori. axes

Epoch

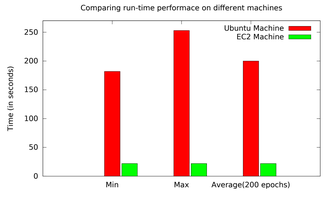
- Train on 300 epochs
- 0.98 split on training set
- Each epoch trained on 49K images, validation on 1K images

NGD parameters: Learning rate $= 10^{-3}$, Decay rate to 10^{-6} , momentum $= 0.9$

Experimental setup

$\sim$10x faster comptuation

- Amazon elastic compute instance (EC2)
  - Nvidia GRID K520 GPU
  - Xeon E5-2670 processes 8 hyperthreads
  - 15 GiB RAM
- OpenBLAS to optimize NumPy
- CUDA drivers with CuDNN library



Kaggle submissions report card

Accuracy for all Kaggle submissions

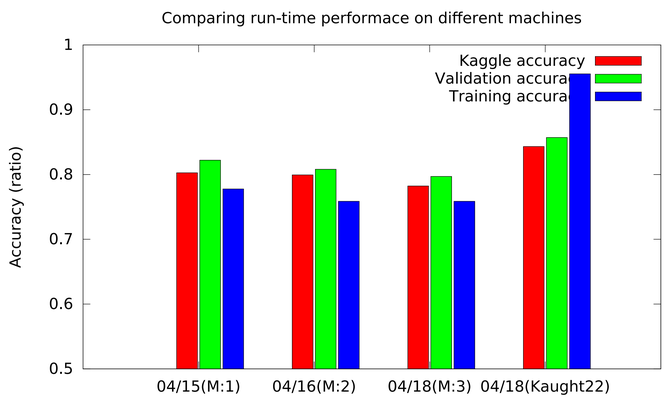


Figure : Trends of accuracy on training data and validation data on Kaught22

Accuracy trends on Kaught22

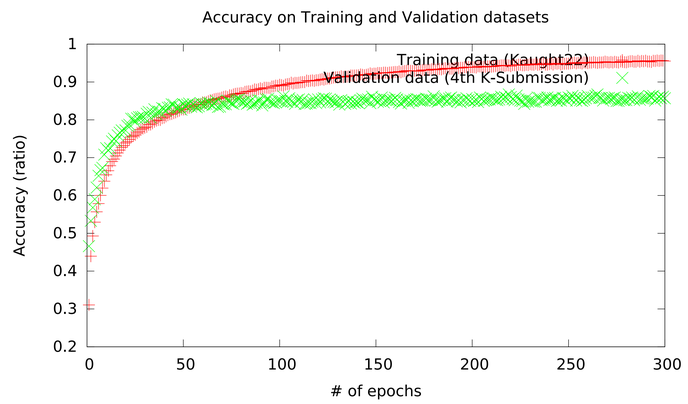


Figure : Trends of accuracy on training data and validation data on Kaught22

Other Kaggle submissions

- No mean shift data preprocessing. Rest data proprocessing identical to Kaught22
- Same epoch configuration as Kaught22, number of epochs may vary
- Same configuration for NGD as Kaught22

04/15 (M:1)

- Identical to Kaught22, aggressive dropout ratio= 0.25 on CONV Blocks
- Trained on 200 epochs

04/16 (M:2)

- Additional CONV Blocks

$$\mathbf{CONV(32, 64, 0.25) - CONV(64, 64, 0.25) - CONV(128, 64, 0.25)-}$$

$$\mathbf{-FC(256, ReLu, 0.5) - FC(10, SoftMax, 0)}$$

- Trained on 200 epochs

04/18 (M:3)

- Identical to Kaught22, meek dropout ratio= 0.125 on CONV Blocks
- Trained on 300 epochs

Conclusion

Experimental setup

- EC2 set up reduces runtime by $\sim$10x

Hyper-parameter tuning

- Adding more layers less beneficial
- Too aggressive and too meek dropout regularization not helpful

Data preprocessing

- Mean shift improves accuracy drastically
- Hori. and vert. flips makes model more sturdy

Source Code

`https://github.com/HarshUpadhyay/COMP540_Project`