

Transfer Learning

The choice of weight initialization in training

Ming-Jia Huang, ICL Div. X, ITRI

LAB03-1

- Please construct the LeNet-5 manually and use MNIST as dataset
- 5 kinds of weight initialization methods
 - ⊙ All weights are zero
 - ⊙ Uniform distribution $(-1, 1)$
 - ⊙ Normal distribution, mean = 0 & standard deviation = 0.1
 - ⊙ Xavier initialization
 - ⊙ Kaiming initialization
- Summary the observation during the training with different initialization method

LeNet-5

- Paper
 - ⊙ Gradient-Based Learning Applied to Document Recognition

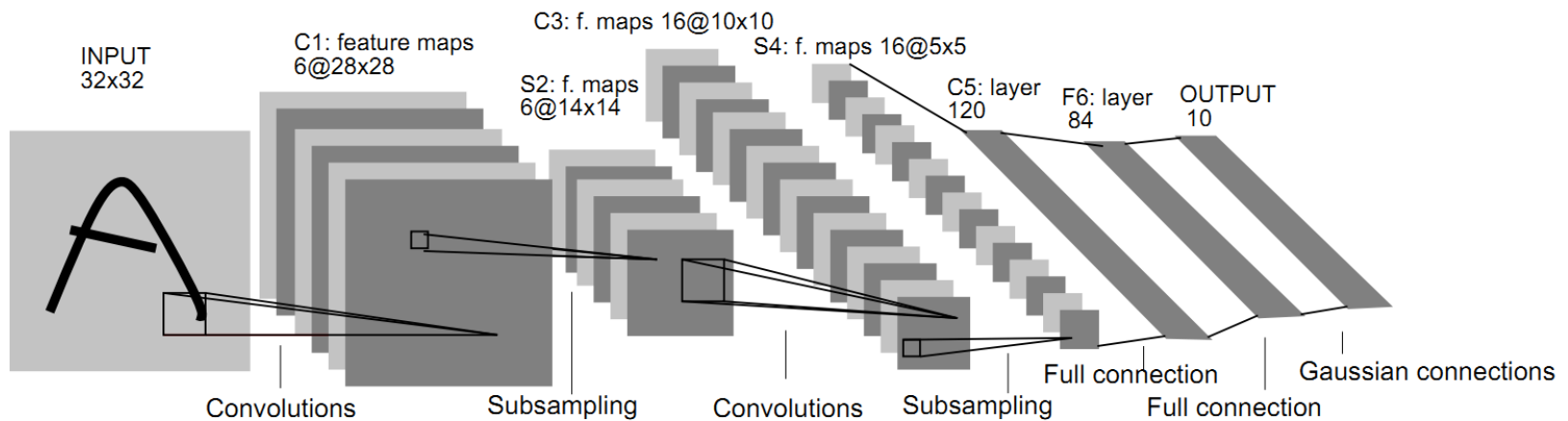


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Hyper Parameters

- Learning Rate: $1e-3$
- Optimizer: ADAM
- Loss Function: Cross Entropy
- Epoch: 1
- Batch Size: 32
- Dataset: MNIST
- Activation Function: ReLU

Xavier & Kaiming initialization

- The basic idea: keep the variance from input to output
- Xavier initialization:
 - ⊙ Understanding the Difficulty of Training Deep Feed-Forward Neural Networks

$$W_i \sim N\left(0, \sqrt{\frac{2}{n_{in} + n_{out}}}\right)$$

- Kaiming initialization
 - ⊙ Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification

$$W_i \sim N\left(0, \sqrt{\frac{2}{(1 + a^2)n^l}}\right)$$

No idea on what's point be observed?

- For more clever in exploration
 - ⊙ Visual the weight distribution during the training
 - ⊙ Plotting the accuracy curve
- Questions
 - ⊙ Why training efficiency has difference if more suitable weight initialization method be chosen?
 - ⊙ Why setting weights to zero will be suffering accuracy stop increasing?

BOOOOOOONUS

- This **BONUS** is **NOT A PART OF TEAMWORK** then you guys can skip it
- All hyper parameters can be modified, except the following:
 - ⊙ Dataset must be MNIST
 - ⊙ Network must be LeNet-5, but activation function is changeable
- The Goal:
 - ⊙ Just for FUN
 - ⊙ Achieve 98.5% accuracy during **1 EPOCH** stably (no lucky)

```
Epoch: 0, Batch: 1600, Avg. Loss: 0.0019649, Accuracy: 0.979
Epoch: 0, Batch: 1650, Avg. Loss: 0.0015159, Accuracy: 0.984
Epoch: 0, Batch: 1700, Avg. Loss: 0.0015733, Accuracy: 0.984
Epoch: 0, Batch: 1750, Avg. Loss: 0.0019723, Accuracy: 0.986
Epoch: 0, Batch: 1800, Avg. Loss: 0.0015585, Accuracy: 0.982
Epoch: 0, Batch: 1850, Avg. Loss: 0.0015130, Accuracy: 0.984
```

It's a lucky, not the real