# Advanced techniques

## Lecture 6

Changho Suh

January 23, 2024

# Hyperparameter search and cross validation

# Outline

1. Hyperparameter search

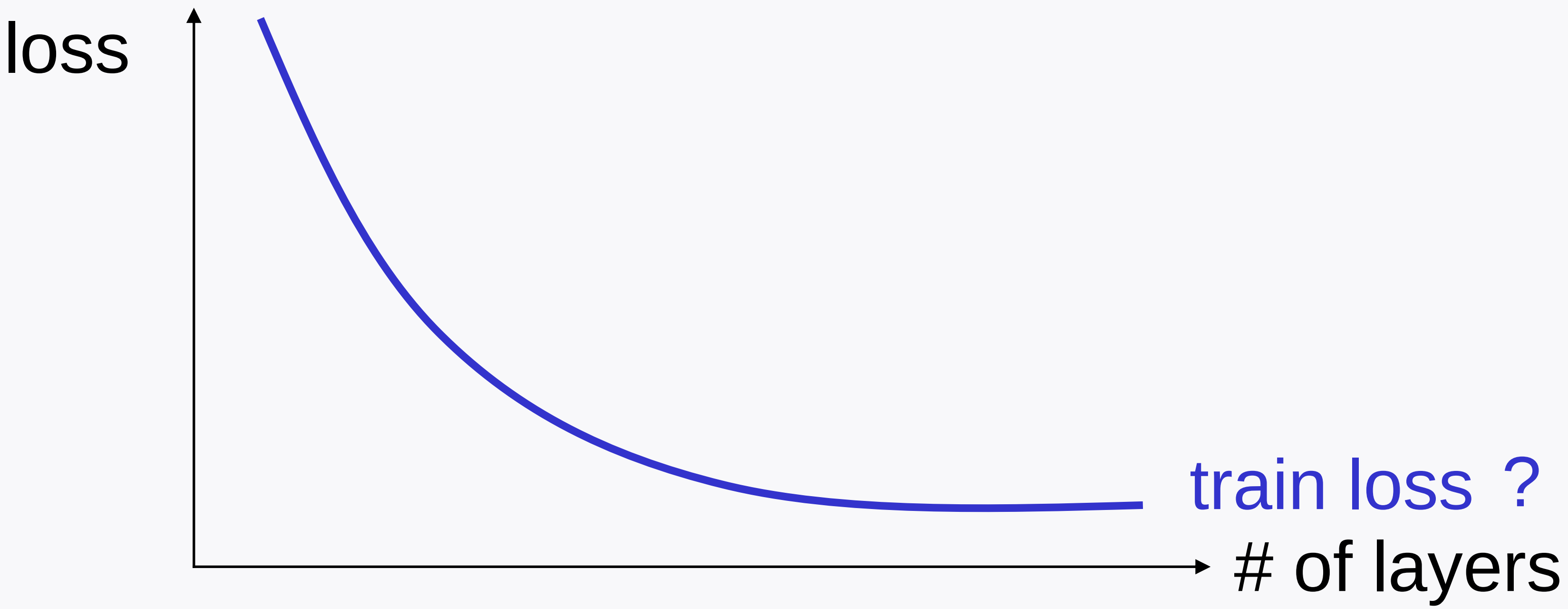
   # $L$ of layers, # $n^{[\ell]}$ of hidden neurons, activation learning rate, betas, batch size, # $T$ of epochs, regularization factor, dropout rate, …

2. Cross validation

# # of layers

Just begin with a **single hidden** layer: $L = 1$

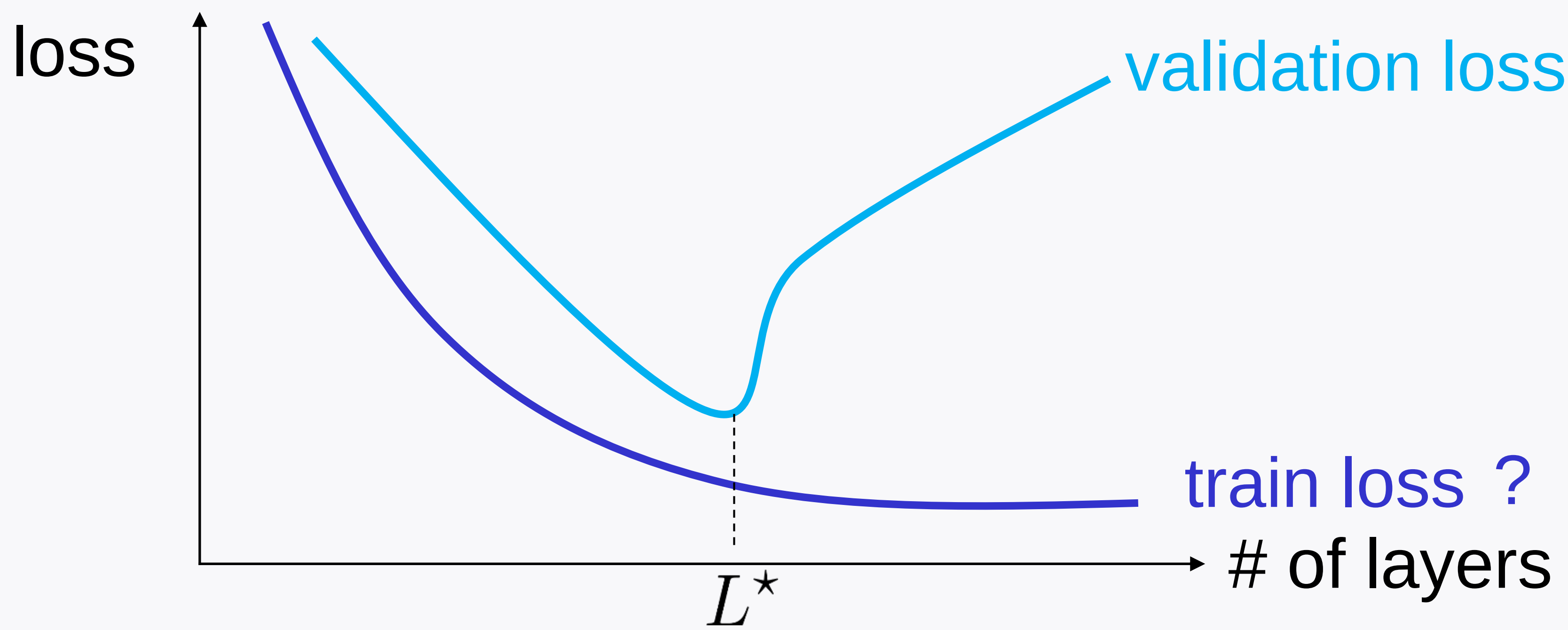
**Gradually (linearly)** ramp up # of hidden layers.



# # of layers

Just begin with a **single hidden** layer: $L = 1$

**Gradually (linearly)** ramp up # of hidden layers.




Stop when overfitting starts.

# # of layers

When increasing *L*:

How to set the number of hidden neurons for all hidden layers?

For the time being:

Set that number around one half of the number of input neurons:

$$n^{[\ell]} = \frac{n}{2}$$

# # of hidden neurons

Two approaches:

1. Fewer neurons for deeper layers

2. Same size for all hidden layers:

   Linearly increase the size until not overfitting.

# Activation functions

A default setup:

Hidden layers: ReLU

Output layer: **Softmax** for classification

**No activation** for regression

# Optimizer

A default use: **Adam**

Default parameters: $(\beta_1, \beta_2) = (0.9, 0.999)$

Two approaches for a choice of the learning rate:

1. Learning rate decaying

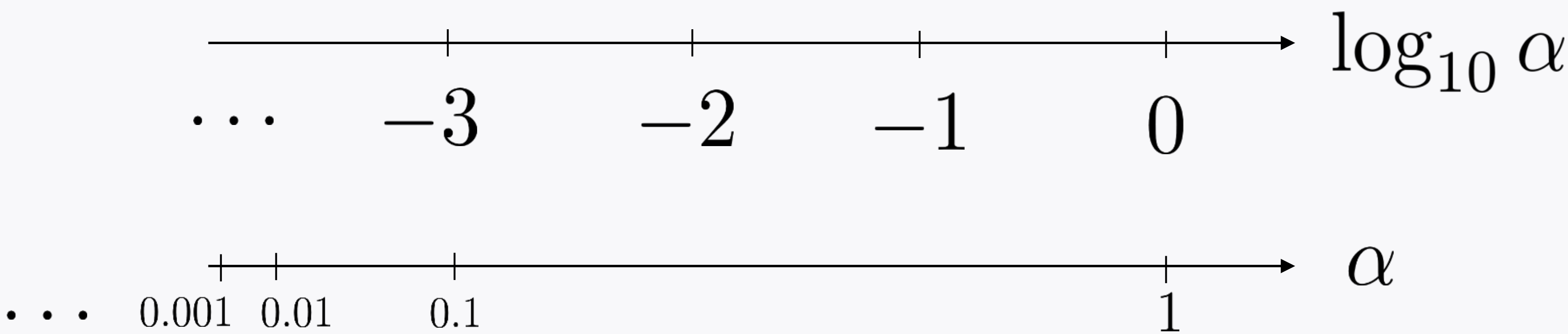
2. Fixed (e.g., $\alpha = 0.001$ )

# How to choose a fixed value of $\alpha$

Do not use a linear-scale grid search.

Try **random** values and **then do a fine search** around the good choices.

Grid scale for the fine search: **Log** scale



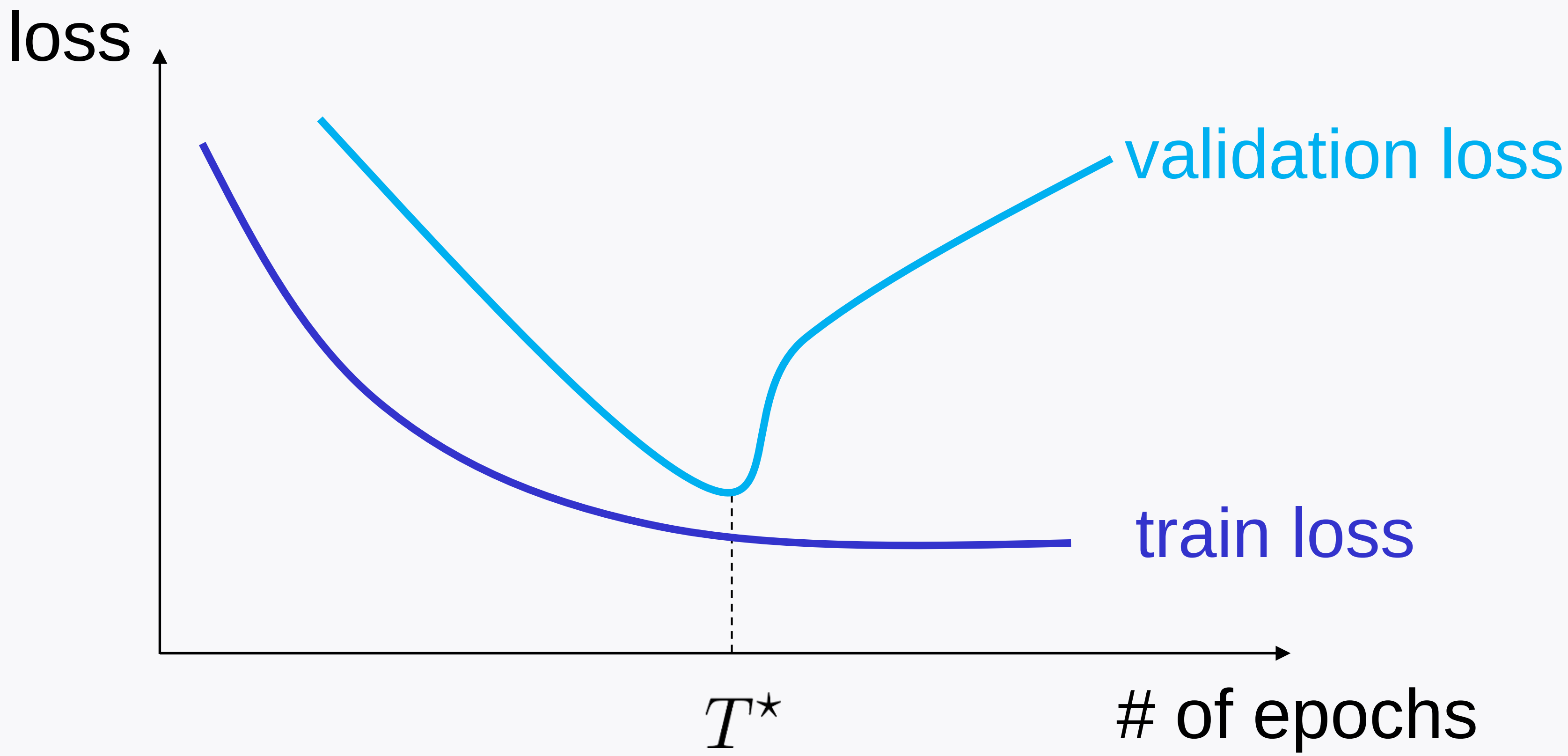
# Batch size

A common choice: **Power of two**.

4, 8, 16, 32, 64, 128, 256

# # of epochs

Choose according to **early stopping**:



# Regularization factor

**Log-scale** search:

# Dropout rate

A typical choice: $p = 0.5$

A good range: $0.2 \leq p \leq 0.8$

# Cross validation

**Purpose:** Obtain **reliable** validation loss via **averaging.**

→ Helps avoid overfitting

Example: 4-fold cross validation

| | | | | |
|---|---|---|---|---|
| val | train | train | train | test |

→ Compute a validation loss, say $\mathsf{val}_1$

Take the 2nd partition for val:

| | | | | |
|---|---|---|---|---|
| train | val | train | train | test |

→ Compute a corresponding loss: $\mathsf{val}_2$

# Cross validation

| | | | | | |
|---|---|---|---|---|---|
| val | train | train | train | test | $\mathrm{val}_1$ |
| train | val | train | train | test | $\mathrm{val}_2$ |
| train | train | val | train | test | $\mathrm{val}_3$ |
| train | train | train | val | test | $\mathrm{val}_4$ |

Take the average over the 4 losses:

$$\text{val loss} = \frac{\mathrm{val}_1 + \mathrm{val}_2 + \mathrm{val}_3 + \mathrm{val}_4}{4}$$

Choose a hyperparameter that minimizes the average loss.

# A final model w.r.t. the best hyperparameter?

| | | | | | |
|---|---|---|---|---|---|
| val | train | train | train | test | $\text{model}_1$ |
| train | val | train | train | test | $\text{model}_2$ |
| train | train | val | train | test | $\text{model}_3$ |
| train | train | train | val | test | $\text{model}_4$ |

Which one to take among the four models?

A final model is the one trained based on:

| | | | | |
|---|---|---|---|---|
| train | train | train | train | test |

# What is next?

One important question:

Can DNNs be specialized?

*CNNs*: Image data

*RNNs*: Text/audio data (language) and
any sequential data

# Outline of Day 3 lectures

Focus on CNNs.

Specifically we will:

1. Investigate how CNNs were developed;
2. Study the two key building blocks:

   Conv layer

   Pooling layer
3. Discuss popular CNN architectures.