

Title: Machine Learning Lecture

Speakers: Mohamed Hibat Allah

Collection: Machine Learning 2023/24

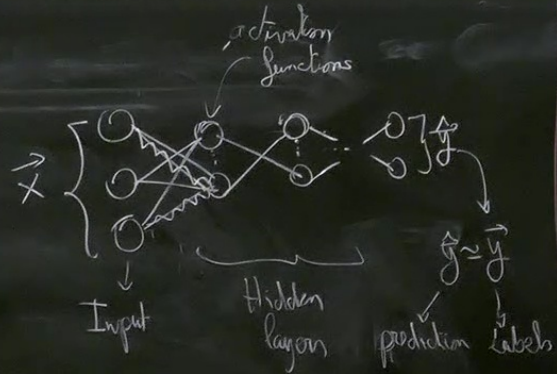
Date: April 15, 2024 - 3:15 PM

URL: <https://pirsa.org/24040056>

Lecture 5

Last time

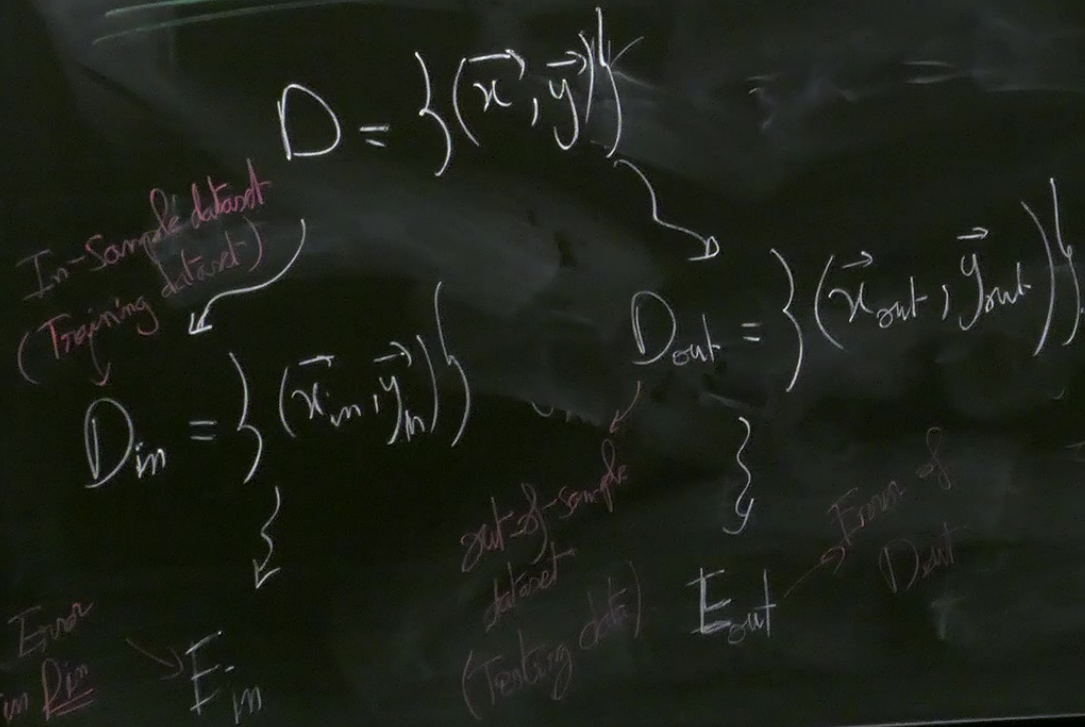
- Feed-forward NNs
- Variants of gradient descent
- Backpropagation
- Demo of overfitting

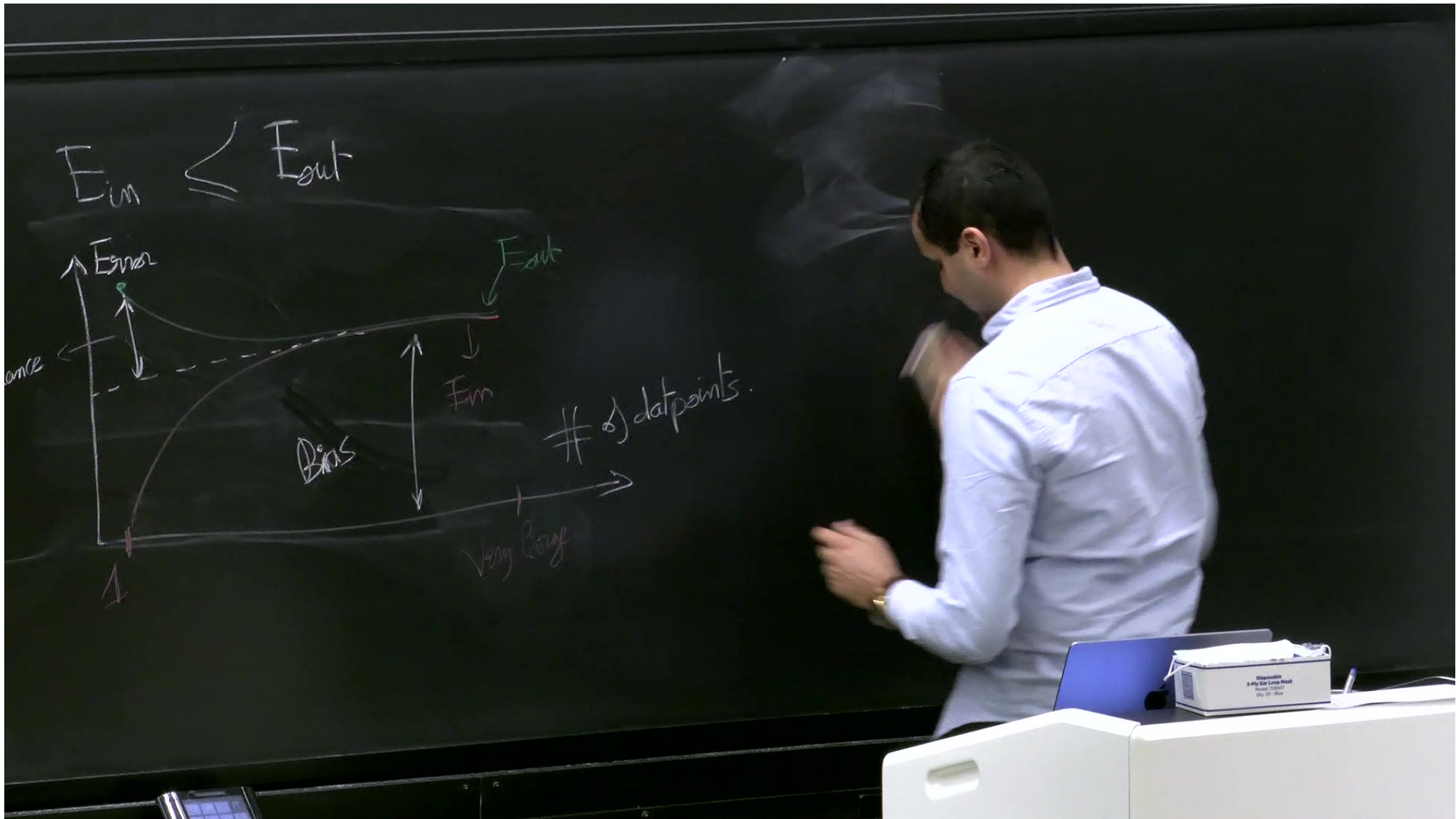


Outline for today

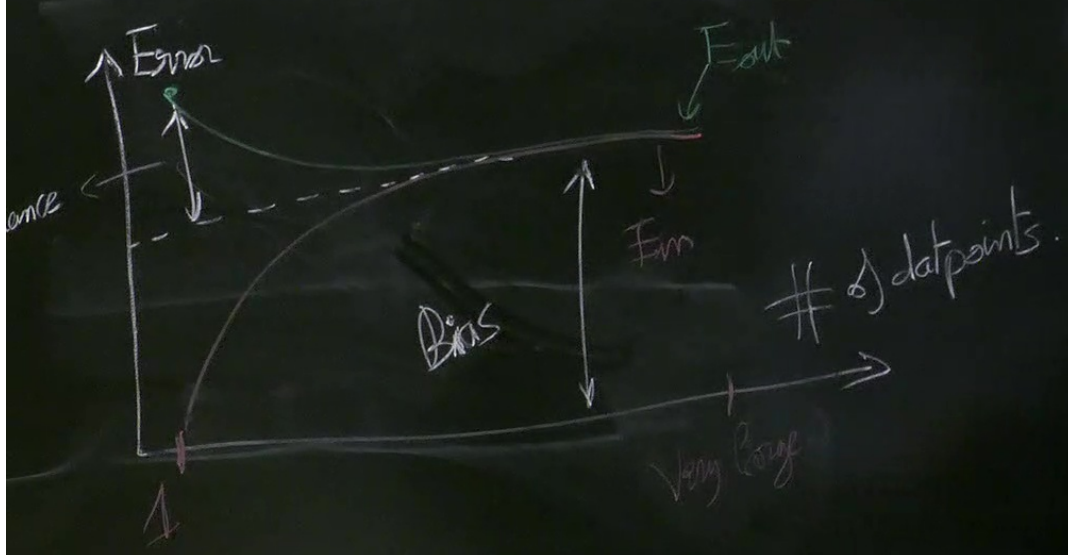
- Bias-Variance tradeoff
- Overfitting
 - ↳ Training, validation, and testing datasets
 - ↳ Methods for preventing overfitting
- Learning phases of mitter (Ising model)

Bias - Variance tradeoff





$$E_{in} \leq E_{out}$$

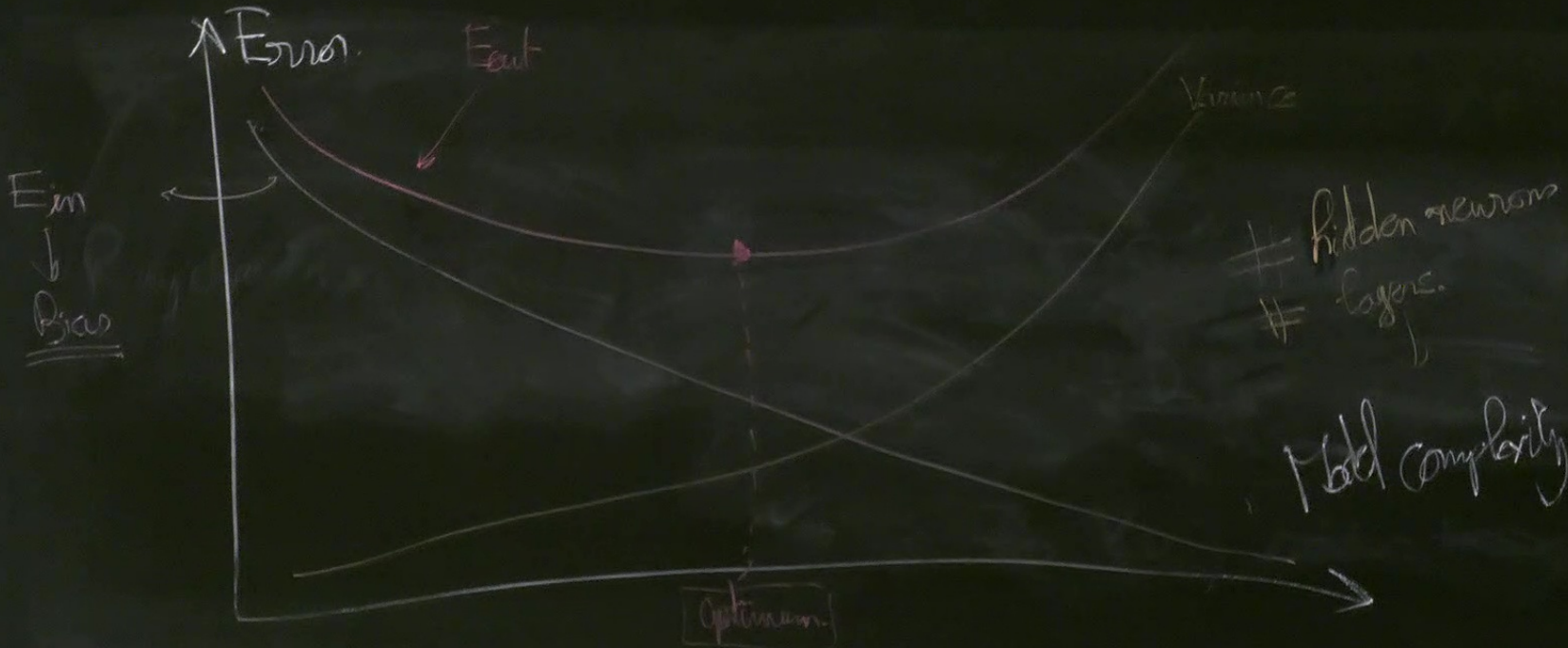


in D_{in} $\rightarrow F_{in}$ (Testing on D_{out})

« Bias » = How powerful to describe your D_{in} dataset
« Variance » = how dependent the fit on the realization of the training data

2010





Firefox File Edit View History Bookmarks Tools Window Help

PSI Portal Polynomials_fitting_illustrat A Neural Network Playgroun A Neural Network Playgroun A Neural Network Playgroun Deeper Playground - Deep L X Deep double descent Deep Double Descent (cross X +

https://colab.research.google.com/drive/1d8019_xyNKakm8axGp8_8RnDLBJ59_ux?authuser=1#scrollTo=znS-PmnZWkk2 133%

Polynomials_fitting_illustration.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

run(d=3)

Legend:
— ground-truth
• samples
— model

RAM
Disk

Connected to Python 3 Google Compute Engine backend

Mac OS dock with various application icons including Safari, Mail, Photos, Calendar, Chrome, Firefox, VS Code, and others.

Firefox File Edit View History Bookmarks Tools Window Help

PSI Portal Polynomials_fitting_illustration A Neural Network Playground A Neural Network Playground A Neural Network Playground Deeper Playground - Deep L Deep double descent Deep Double Descent (cross

https://colab.research.google.com/drive/1d8019_xyNKakm8axGp8_8RnDLBJ59_ux?authuser=1#scrollTo=znS-PmnZWkk2 133%

Polynomials_fitting_illustration.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

run($h=3$)

Legend:
— ground-truth
● samples
— model

Connected to Python 3 Google Compute Engine backend

Mac OS X dock with various application icons including Safari, Mail, Photos, Calendar, Chrome, Firefox, VS Code, and others.

Firefox File Edit View History Bookmarks Tools Window Help

PSI Portal Polynomials_fitting_illustration A Neural Network Playground A Neural Network Playground A Neural Network Playground Deeper Playground - Deep L Deep double descent Deep Double Descent (cross

https://colab.research.google.com/drive/1d8019_xyNKakm8axGp8_8RnDLBJ59_ux?authuser=1#scrollTo=gaRve9gA2Tzq 133%

Polynomials_fitting_illustration.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

run(d=1000)

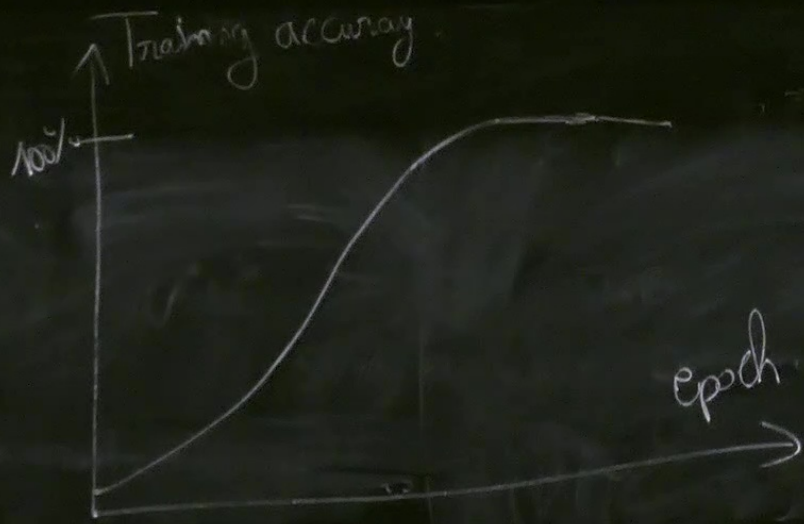
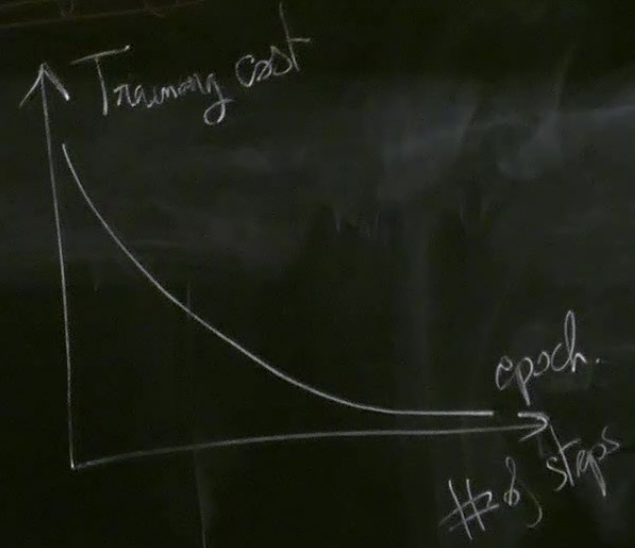
RAM Disk

ground-truth
samples
model

Connected to Python 3 Google Compute Engine backend

Mac OS dock with various application icons including Safari, Chrome, and Zoom.

Overfitting



Underfitting

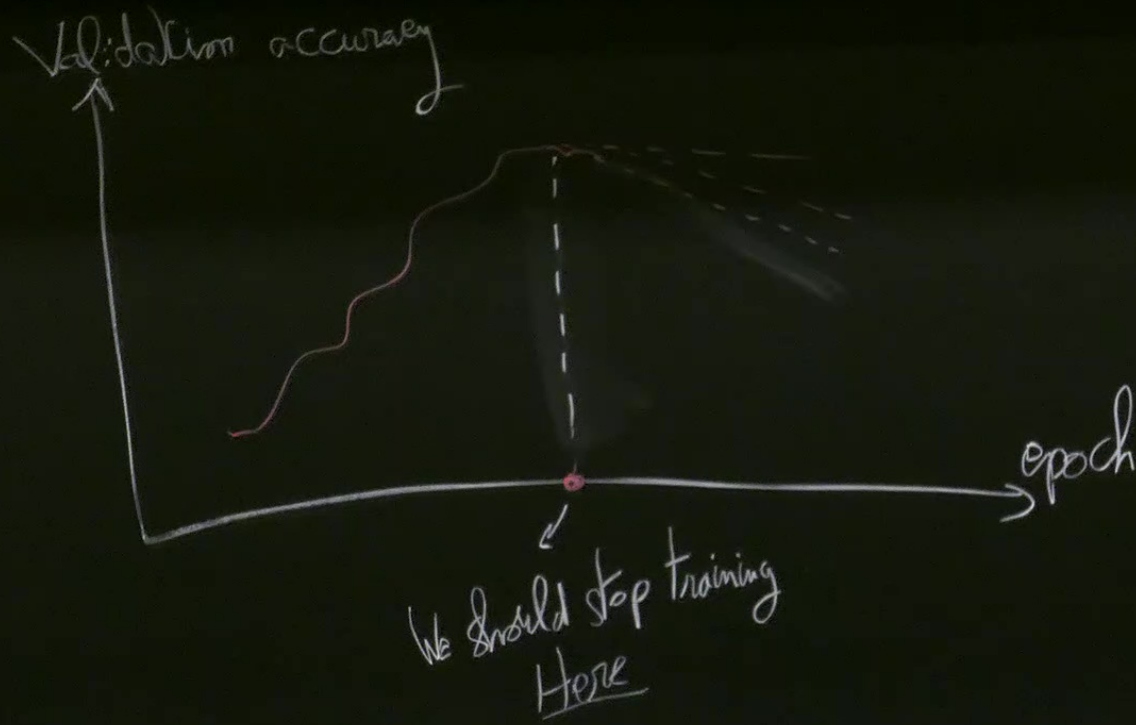
Overfitting

→ Training data: D_{in} → 70% of D

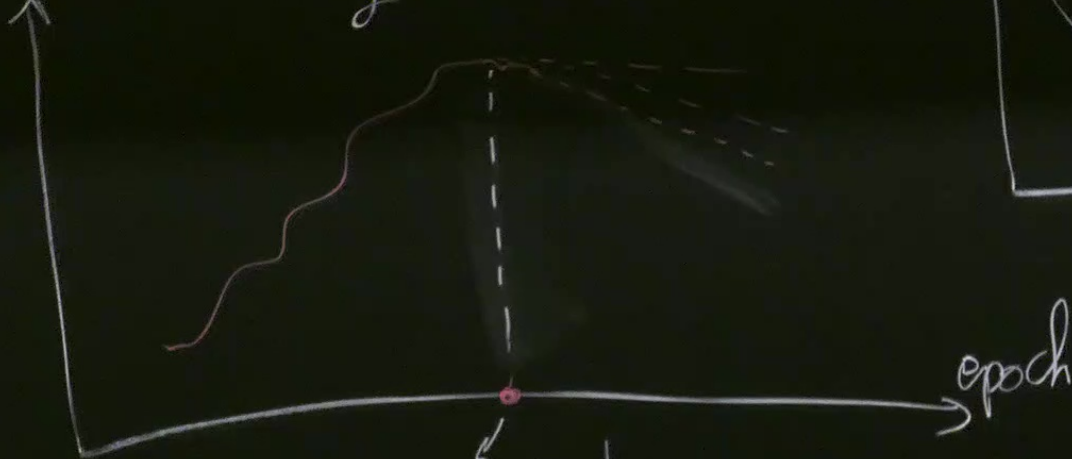
↔ Validation data: → 20% of D

↔ Testing data: → 10% of D

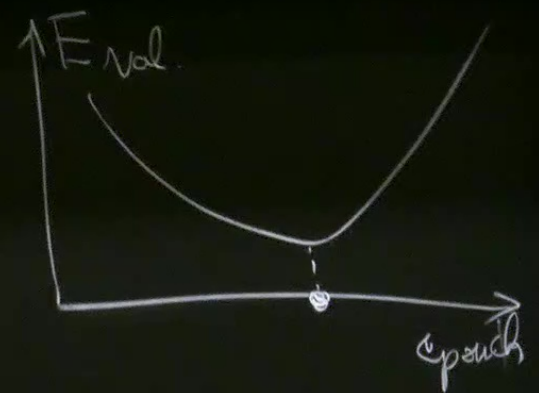
↓
Data



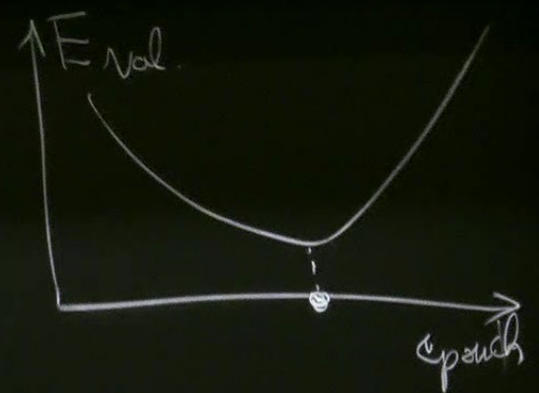
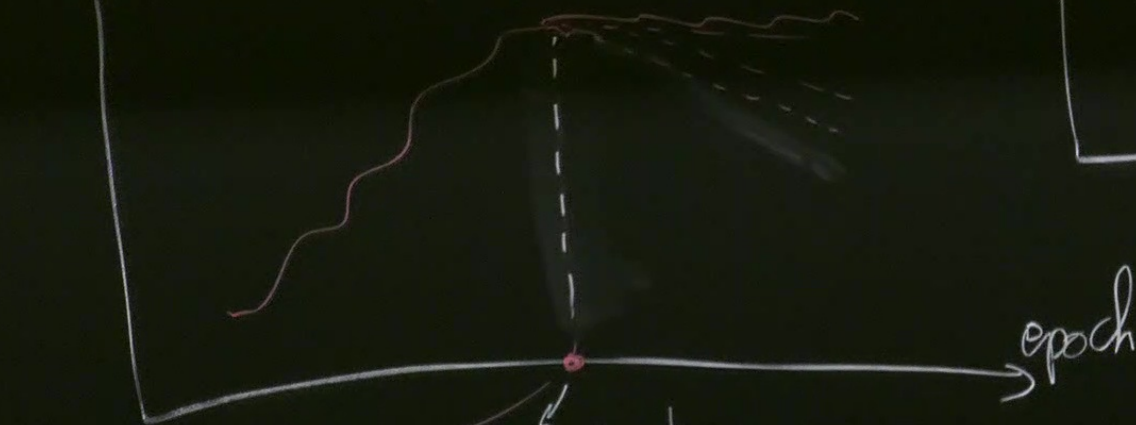
Validation accuracy



We should stop training
Here



Validation accuracy



We should stop training Here

Test accuracy

E_test

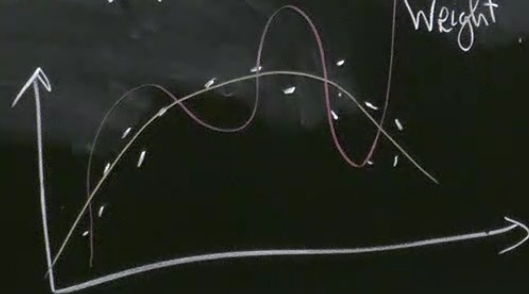
① L1 or L2 regularization \rightarrow regularization hyperparameter

$$C_{L1} = C_{\text{int}} + \lambda \sum_{\text{Weights}} |W_{ij}^{(p)}|$$

$$C_{L2} = C_{\text{int}} + \lambda \sum_{\text{Weight}} |W_{ij}^{(p)}|^2$$

Small \uparrow

$$P(x) = \lambda_4 x^4 + \lambda_3 x^3 + \lambda_2 x^2 + \lambda_1 x + \lambda_0$$



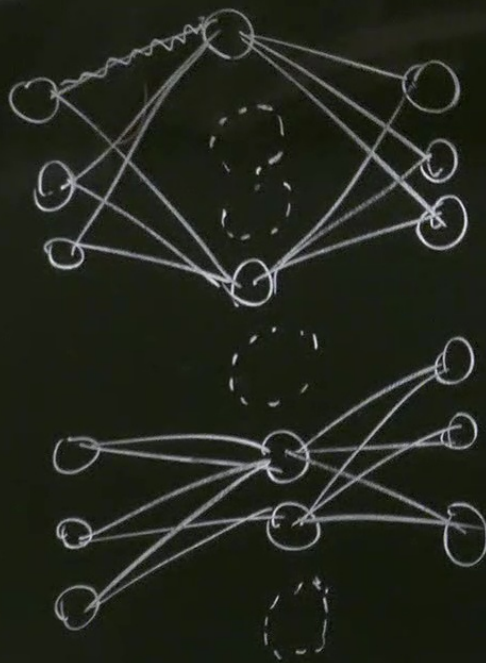
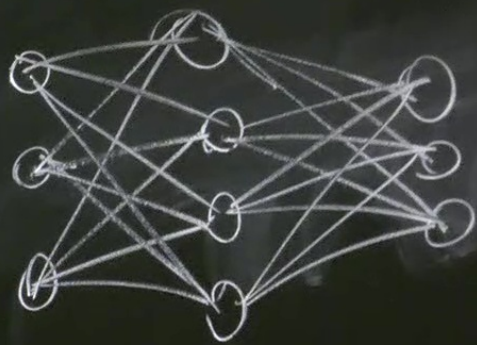
λ too large \rightarrow underfitting.

λ too small \rightarrow overfitting.

Underfitting

Overfitting

② Dropout



Epoch 000,173 Learning rate 0.03 Activation Tanh Regularization None Regularization rate 0 Problem type Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 20%

Noise: 35

Batch size: 10

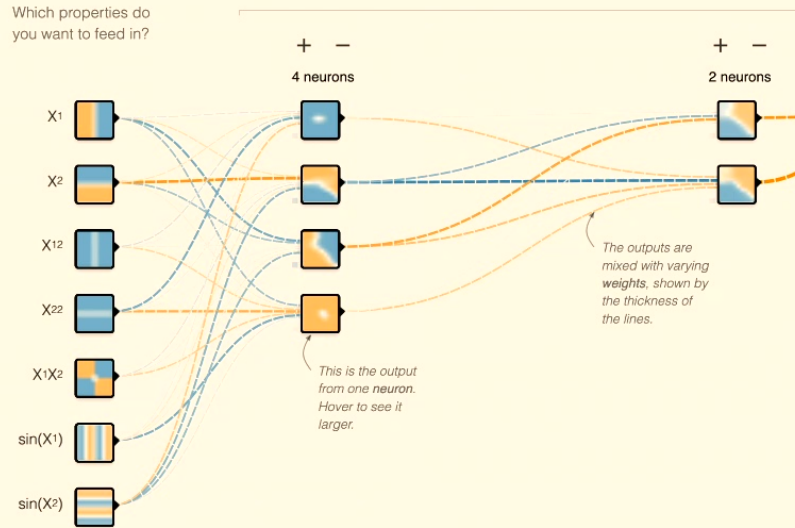
REGENERATE

FEATURES

Which properties do you want to feed in?

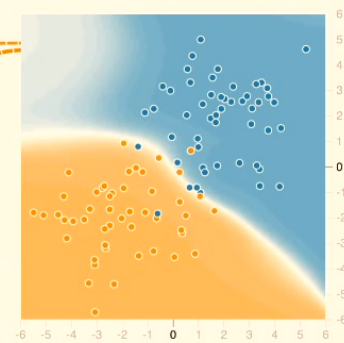
- X1
- X2
- X1²
- X2²
- X1X2
- sin(X1)
- sin(X2)

2 HIDDEN LAYERS



OUTPUT

Test loss 0.082
Training loss 0.076



Colors shows data, neuron and weight values. -1 0 1

Show test data Discretize output

Don't Worry, You Can't Break It. We Promise.

Epoch: 000,000
Learning rate: 0.03
Activation: Tanh
Regularization: L2
Regularization rate: 0.1
Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 20%



Noise: 35



Batch size: 10



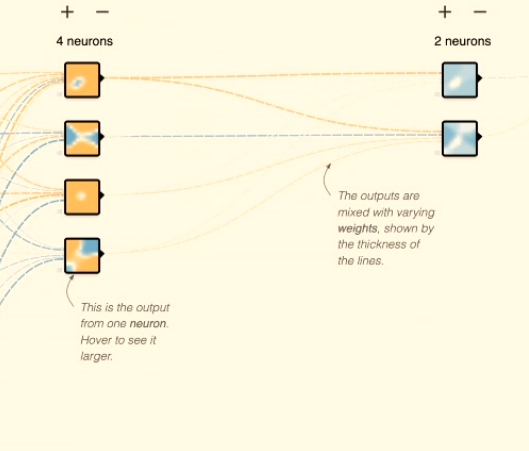
REGENERATE

FEATURES

Which properties do you want to feed in?

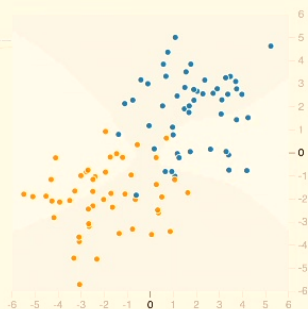
- X1
- X2
- X1²
- X2²
- X1X2
- sin(X¹)
- sin(X²)

2 HIDDEN LAYERS



OUTPUT

Test loss 0.502
Training loss 0.493



Colors shows data, neuron and weight values. Legend: -1 (red), 0 (white), 1 (blue)

Show test data Discretize output

Firefox File Edit View History Bookmarks Tools Window Help

PSI Portal | Polynomials_fitting_illustrat | A Neural Network Playgroun | A Neural Network Playgroun | A Neural Network Playgroun | Deeper Playground - Deep L | Deep double descent | Deep Double Descent (cross | + | -

https://playground.tensorflow.org/#activation=tanh®ularization=L1&batchSize=10&dataset=gauss®Dataset=reg-plane&learningRate=0.03®ularizationRate=0.01&noise=35&networkSI

Epoch: 000,484 | Learning rate: 0.03 | Activation: Tanh | Regularization: L1 | Regularization rate: 0.01 | Problem type: Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 20%

Noise: 35

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X1, X2, X1², X2², X1X2, sin(X1), sin(X2)

2 HIDDEN LAYERS

4 neurons | 2 neurons

Click anywhere to edit. Weight is 0.0.

The outputs are mixed with varying weights, shown by the thickness of the lines.

This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.078
Training loss 0.085

Colors shows data, neuron and weight values.

Show test data Discretize output

Um, What Is a Neural Network?

Epoch: 000,601
 Learning rate: 0.03
 Activation: Tanh
 Regularization: L2
 Regularization rate: 0
 Problem type: Classification

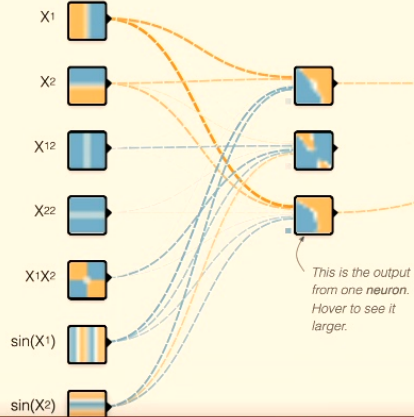
Animation speed: 100%
 Dropout: Drop 50%
 Momentum: 0
 Layerwise gradient normalization: p = 0 (None)
 Learning rate autotuning: None
 Prevent loss increases: No

DATA
 Which dataset do you want to use?



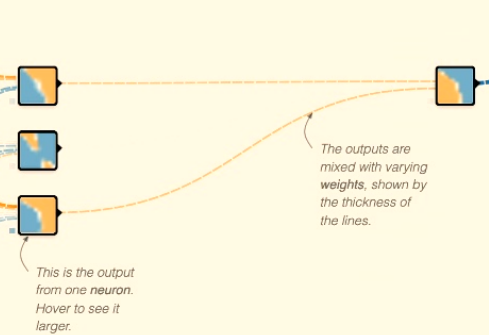
Ratio of training to test data: 20%
 Noise: 35
 Batch size: 10

FEATURES
 Which properties do you want to feed in?



+ - 2 HIDDEN LAYERS

+ - 4 neurons + - 2 neurons



OUTPUT

