Deep Learning 101

Convolutional Neural Network (CNN)

Schedule

week	Date	Topic
9	10.27	Environment setup, python, Jupyter, PyCharm, TensorFlow, & regression
10	11.03	Training and testing
11	11.11	CNN
12	11.18	RNN
13	11.24	Autoencoder & GAN

Today's Class

- Recap
- What is CNN?
  - What is convolution?
  - Stride
  - Padding
  - Max pooling
  - Dropout
- Confusion metrics
- Lab time

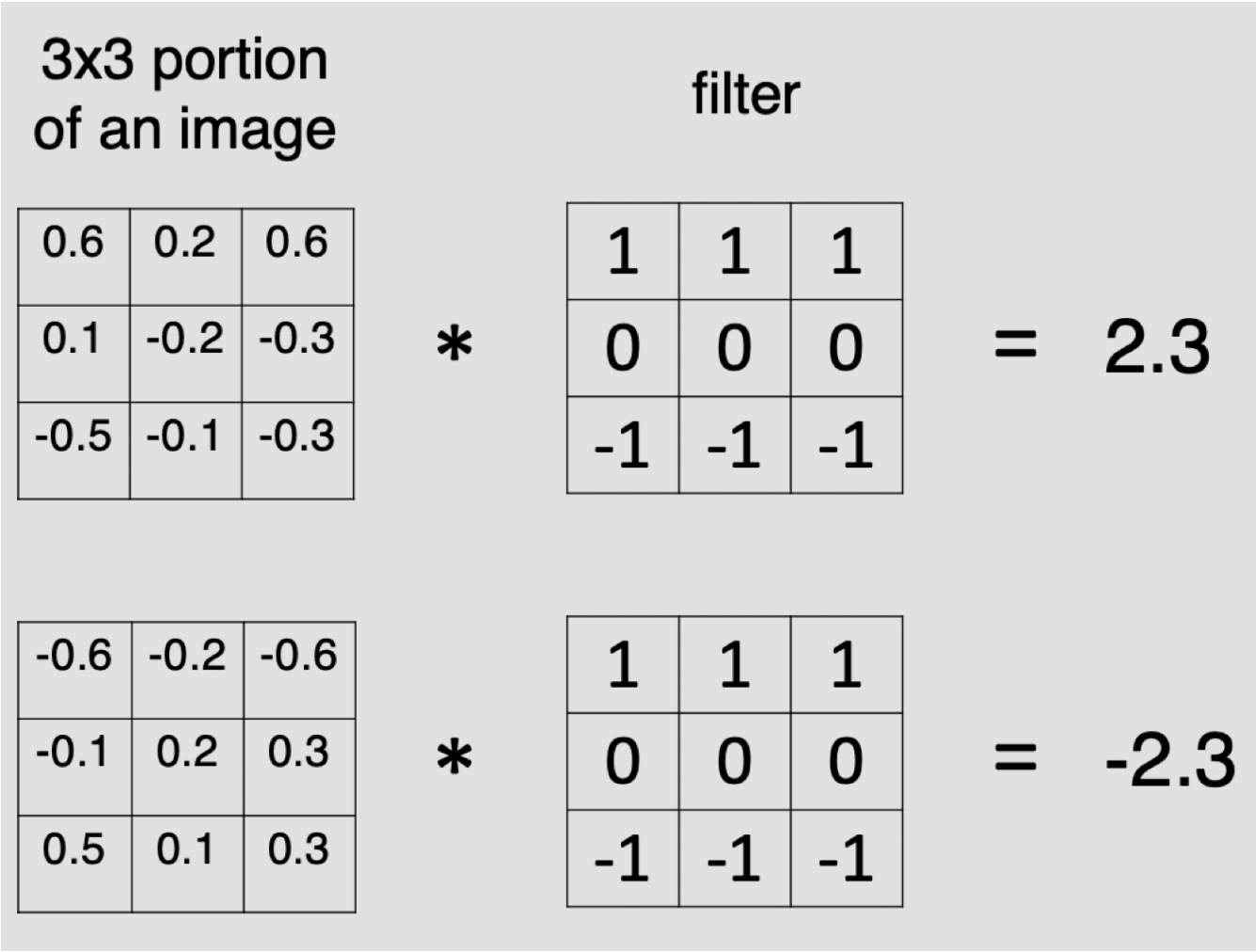
Recap – neural network

- Neural network as a function
  - y = f(x)
- Perceptron
  - Y = WX + b
  - Two inputs: x1, x2
  - One output: y
  - Linear regression
- XOR problem
  - Linear regression can’t solve the XOR problem
  - Require multivariate regression

Recap - Training

- What it take to train a neural network:
  - Hypothesis: H = WX + b
  - Activation function: Sigmoid, tanh, ReLU, LeakyReLU, Softmax, etc.
  - Cost function: MSE, cross entropy, etc.
  - Gradient descent: backpropagation
- Training a neural network is basically the problem of minimizing the cost function: minimize cost(W, b)
- Gradient descent is the most popular optimizer.
- Training a neural network is NOT easy!
  - Finding hyperparameters, random initial weights, local minima, vanishing/exploding gradients, overfitting/underfitting, etc.

How does convolutions work?



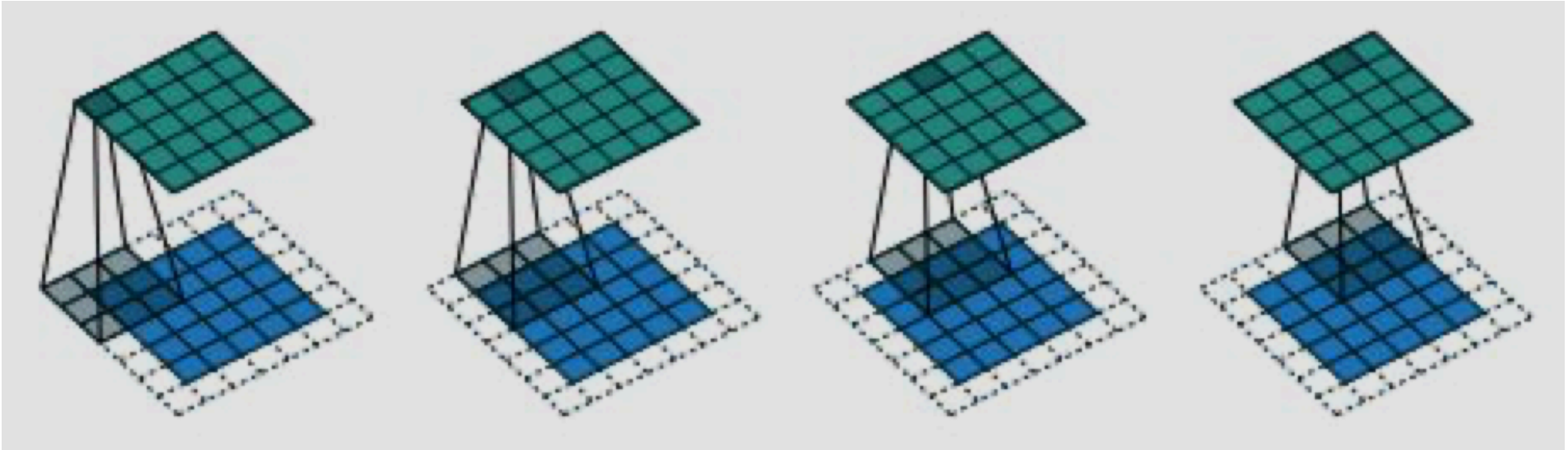
* Generative Deep Learning (David Foster)

What is involved in convolution

- Stride: step size
- Padding: putting zeroes around the outer edge of the input data. “same” means the output size will be the same as the input size when stride = 1.
- Kernel: the filter that extracts features
- Max pooling: means pooling the maximum value from the filtered feature map. The result is a down-sampling image (reduced dimensionality)
- Batch normalization: normalizing
- Dropout: regularization technique to prevent overfitting

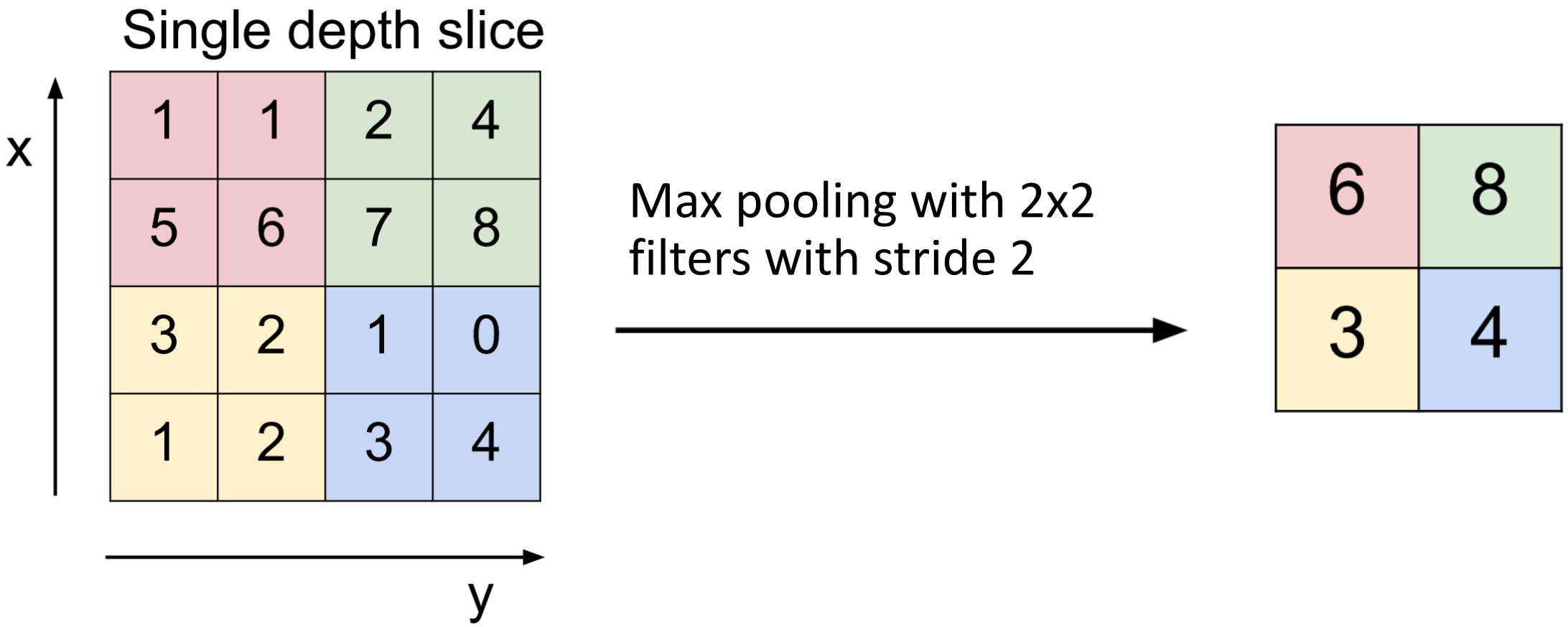
Convolution in action

- stride = 1, padding = “same”



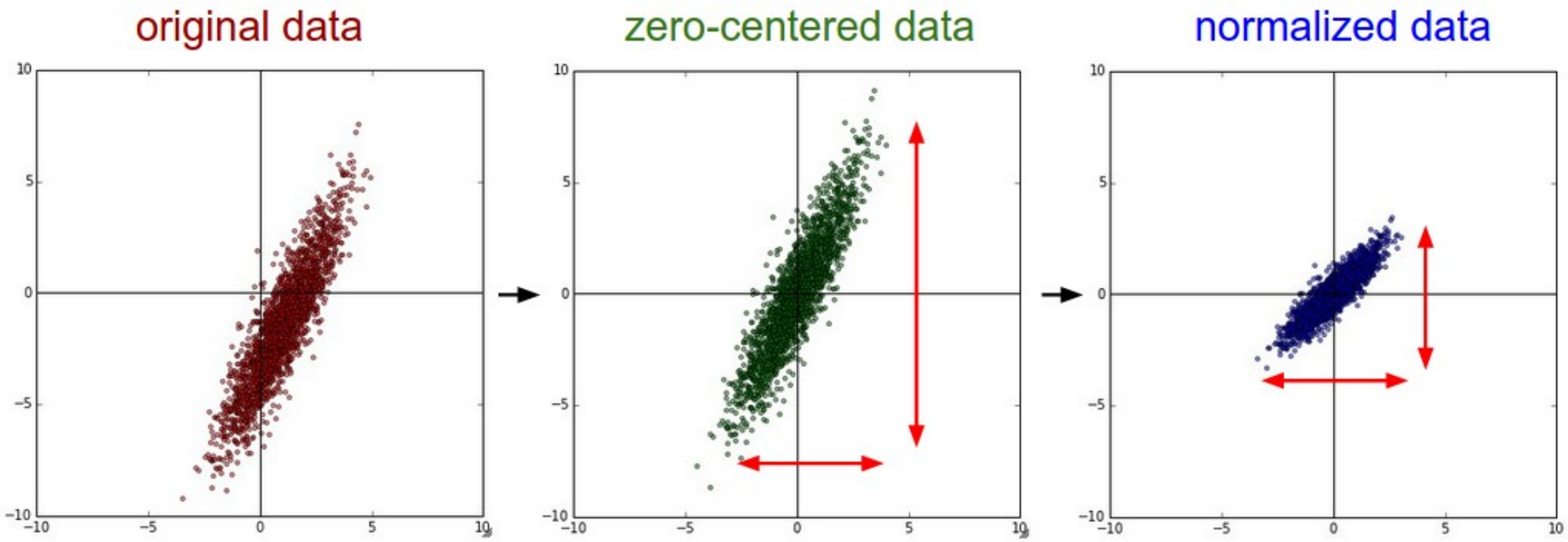
- What will happen if there is no padding?

Max pooling

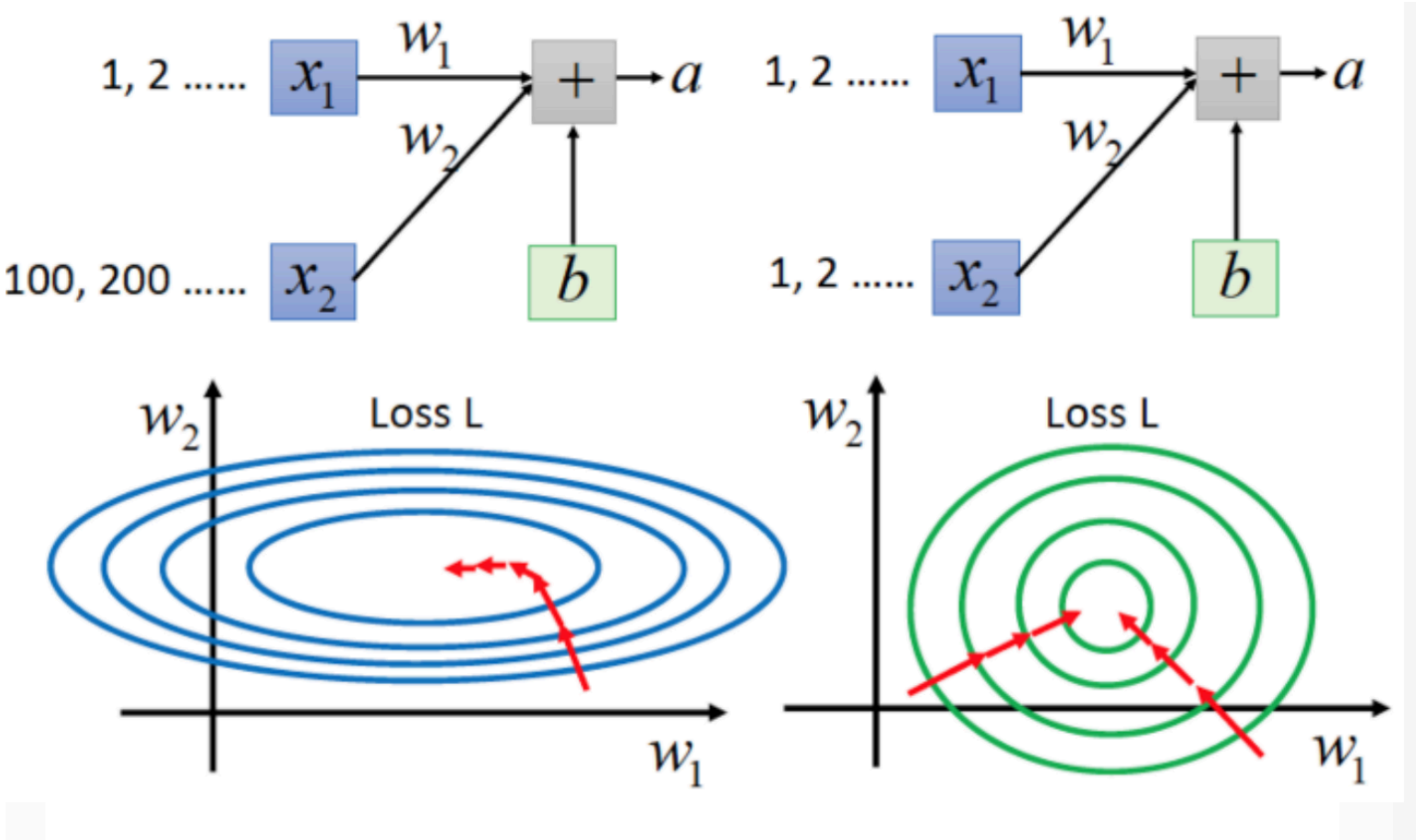

Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
Max pooling with 2x2 filters with stride 2
6 8
3 4

What normalization means

- Zero-centered normalized:

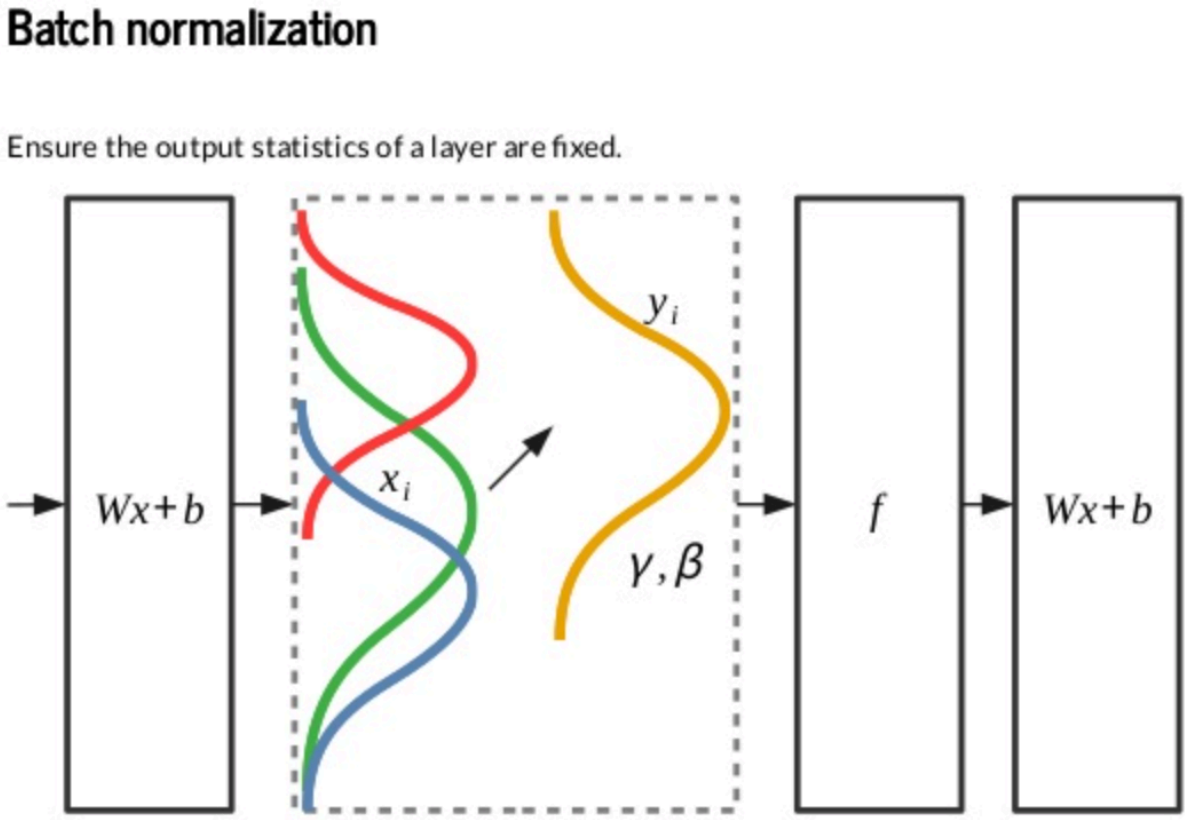


Batch normalization in Deep Learning (intuition)



Batch normalization

- To standardize the input: to mitigate the ”covariate shift” problems



https://www.srose.biz/wp-content/uploads/2020/08/Deep-Learning-Performance-Part-3-Batch-Normalization-Dropout-Noise.html

Batch normalization in action

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

https://www.jeremyjordan.me/batch-normalization/

- normalize the activations of the previous layer for each batch
- apply a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

Visualization Tools

- Convolution Visualizer
- CNN Explainer

Confusion matrix

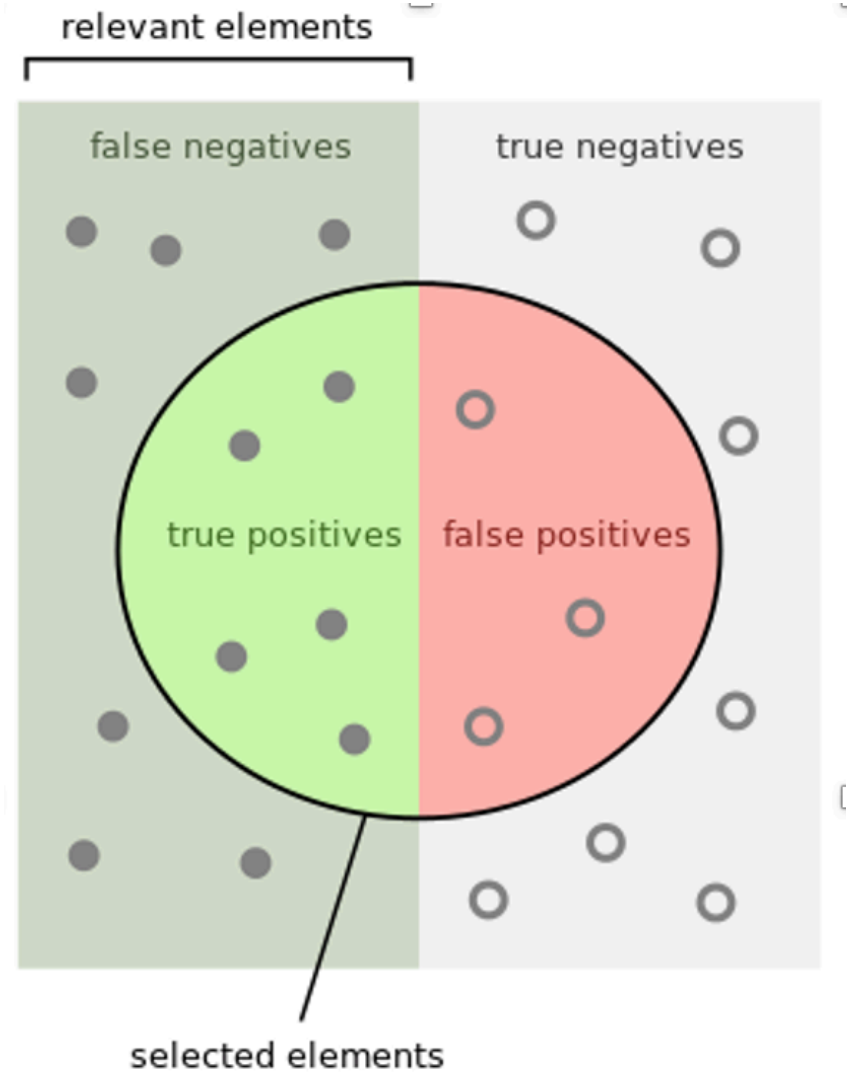
		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

- Are you confused?

Actual example: COVID-19

		Actual	
		Yes	No
Predicted	Yes	True Positive	False Positive
	No	False Negative	True Negative

- Which is worse? False Negative or False Positive?
- What if the problem is different? e.g., spam filter.



How many selected items are relevant?

$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{green}}$$

Metrics: accuracy, precision, and recall

$$\text{Accuracy} = \frac{\text{True Positives + True Negatives}}{\text{All Samples}}$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Metrics

- Problems with accuracy: imbalanced classification problems
- Precision: what proportion of positive predictions was actually correct?
  - (TP) / (TP+FP)
- Recall: what proportion of actual positives was predicted correctly?
  - (TP) / (TP+FN)
-

Lab time

- To clone: from your terminal
  - >git clone https://github.com/changsin/DeepLearning-101.git
- Or use google colab to point to the git hub repository
- Git is an open source version control system
  - Github is a host service using git.