











**☆** Key Points

neural networks.

convolutional layer.



**Batch normalization layers** 

introduce normalization within

Batch normalization is usually

function layer to the output of a

applied before the transition









Course Shortcuts

> Student Lounge

> Q&A

> Explore a Neural Network

 Use Convolutional Neural **Networks for Image Analysis** 

- Module Introduction: Use Convolutional Neural Networks for Image Analysis
- Neural Networks for Images
- Formalize Convolution
- **Review Convolutional** Networks
- Define the Convolutional Operation
- **Explore Image Convolution**
- Identify Patterns in Images
- Explore Pooling Layers
- Visualize a Convolutional Network in Action
- Stabilize Training with **Batch Normalization**



- Preserve Information with Residual and Dense <u>Networks</u>
- ConvNet Cheat Sheet
- PyTorch Tutorial
- Implement a Convolutional Network with PyTorch
- Module Wrap-up: Use **Convolutional Neural** Networks for Image <u>Analysis</u>
- > Analyze Sequence Data with **Neural Sequence Models**

## > Course Resources

Deep Learning and Neural Networks >

## Formalize Batch Normalization

In machine learning, it is common to normalize the input data before we apply any learning.

That is, for each dimension i of the input, we calculate its mean and standard deviation, and we normalize that dimension with  $[\mathbf{x}]_i \leftarrow \frac{[\mathbf{x}]_i - \mu_i}{\sigma_i}$ .

This normalization step ensures that each feature dimension has mean 0 and standard deviation 1, a property that makes training easier. If we are training a linear model using gradient descent, then we might need different learning rates for different dimensions if each dimension is not normalized to the same scale. For neural networks, it has long been

established that feature normalization of the input data can improve the convergence of the network.

Batch normalization layers essentially introduce normalization within neural networks. The procedure is as follows: for each minibatch, we calculate the mean and standard deviation of each dimension of the intermediate activations of the neural networks and normalize it accordingly. The normalized dimensions are multiplied by a learned parameter  $\gamma$  and offset by a learned parameter eta.

Batch normalization re-normalizes activations based on current mini-batch

$$[a]_i \leftarrow \gamma_i rac{[a]_i - \mu_i}{\sqrt{\sigma_i^2 + arepsilon}} + eta_i$$

where  $\mu_i$  is the mean activation in mini-batch and  $\sigma_i^2$  is the variance in mini-batch for dimension i. The parameters that are learned are  $\gamma_i$  and  $eta_i$ . We add a little arepsilon to avoid division-by-zero errors.

Batch normalization is usually applied to the output of a convolutional layer and before the transition function layer. Having this normalization allows us to use larger learning rates and makes training converge faster.

During testing, we use the mean and variance computed from the full training set at the end of training.

◆ Previous

Next ▶