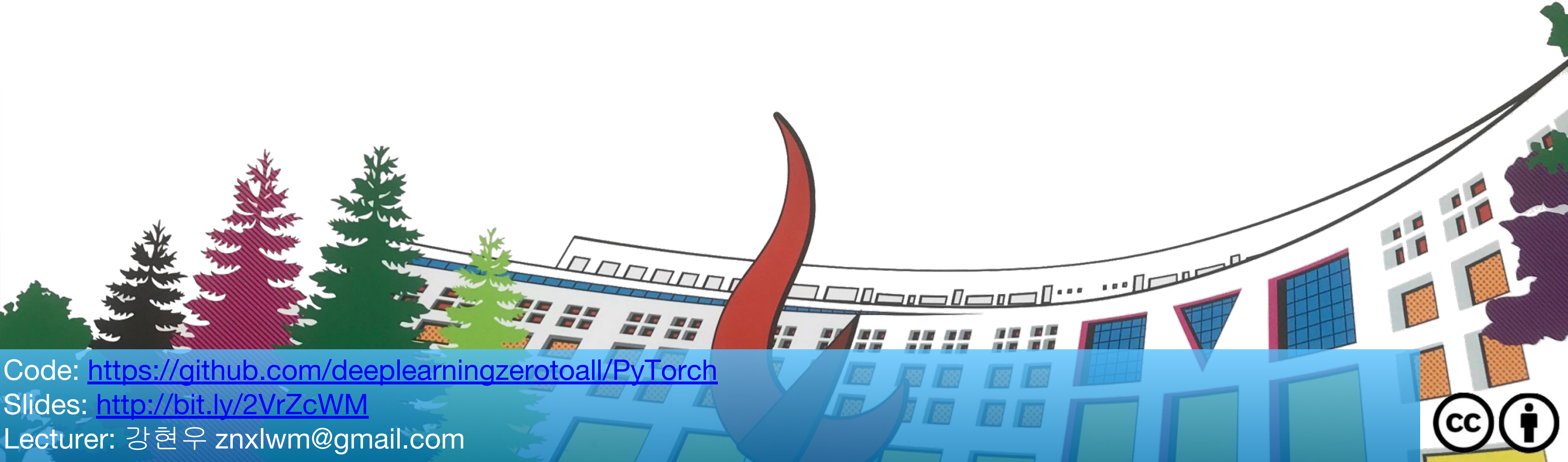
ML/DL for Everyone Season2

Dropout



Code: https://github.com/deeplearningzerotoall/PyTorch

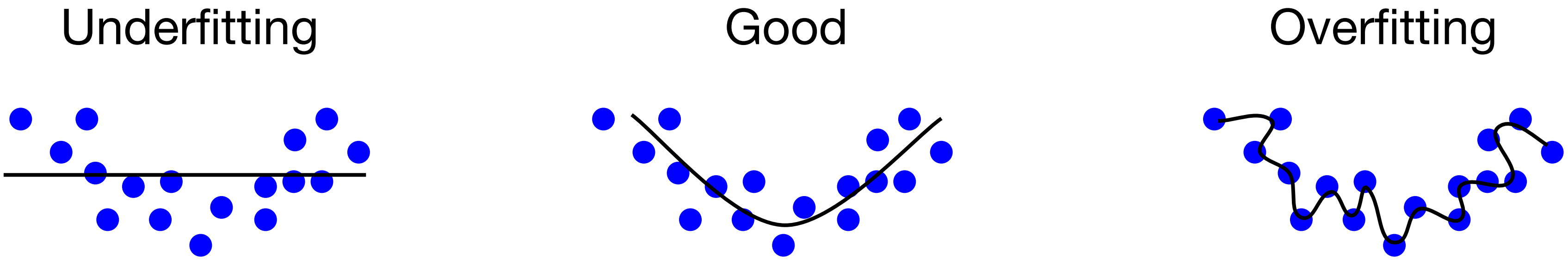
Slides: http://bit.ly/2VrZcWM

Lecturer: 강현우 znxlwm@gmail.com

Dropout

- Overfitting
- Dropout
- Code: mnist_nn_dropout

Overfitting

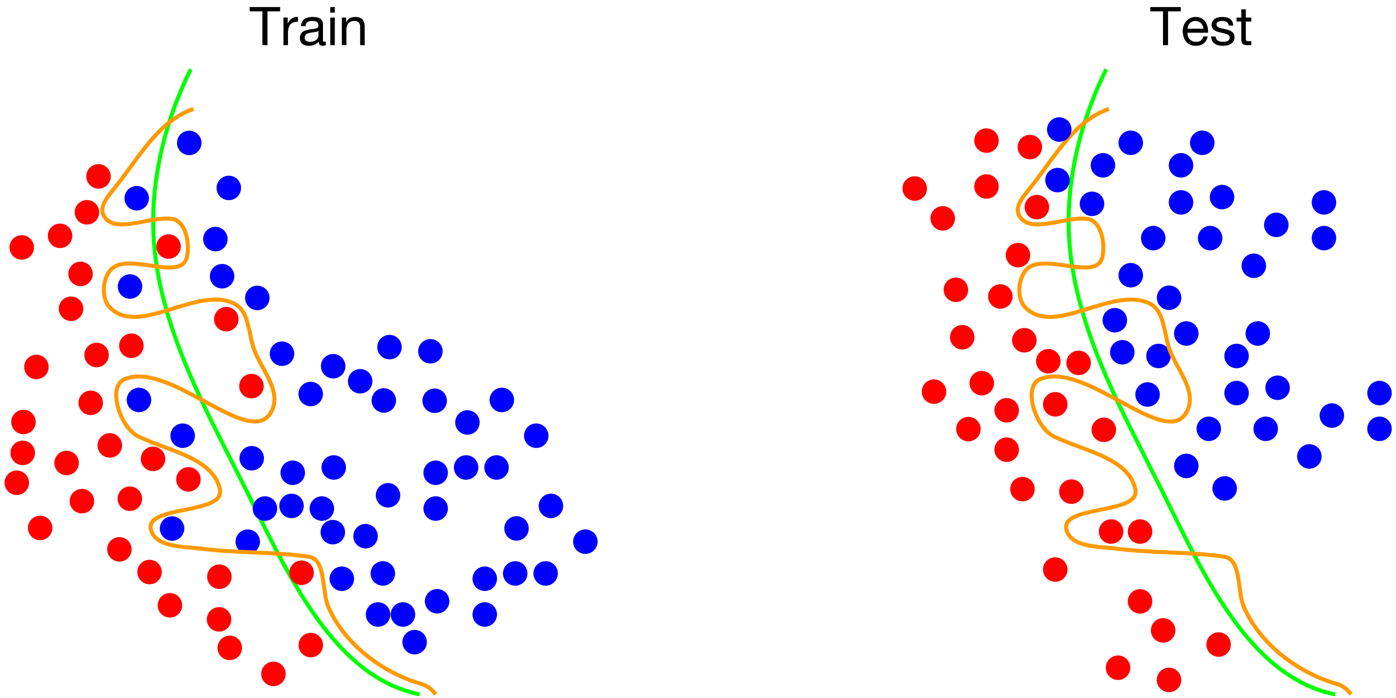


Underfitting

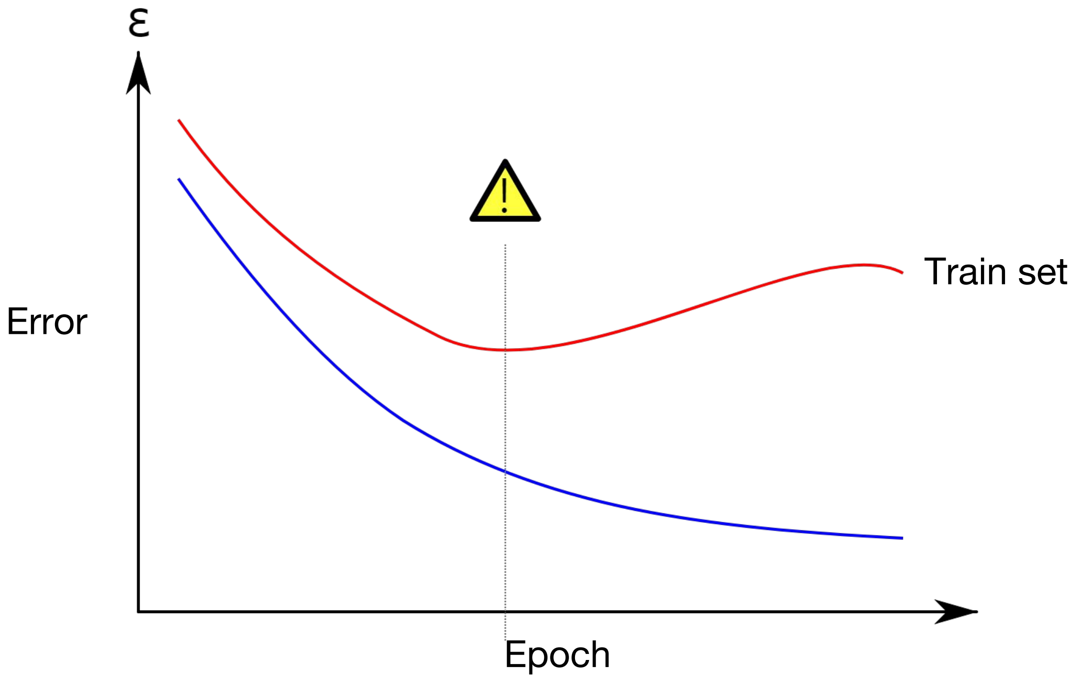
Good

Overfitting

Overfitting



Overfitting



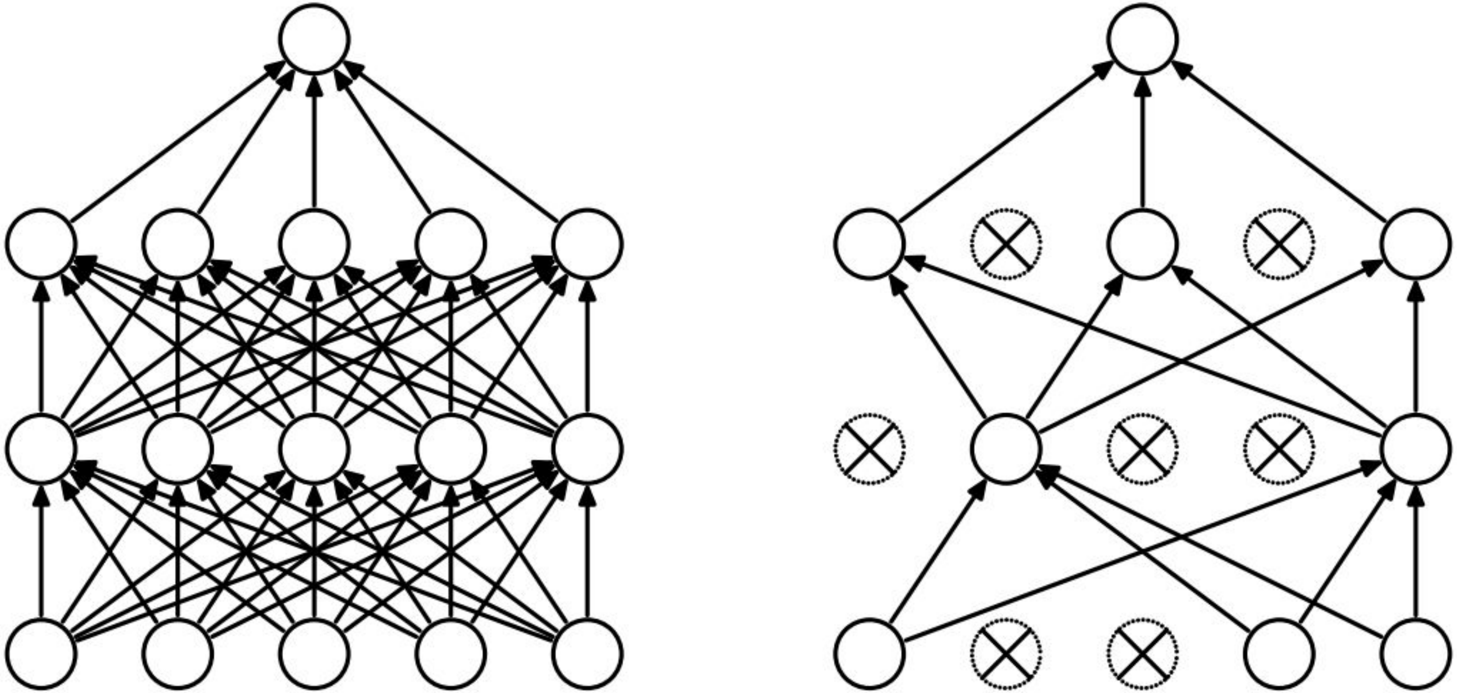
- Very high accuracy on the training dataset (e.g., 0.99)
- Poor accuracy on the test dataset (e.g., 0.85)

https://commons.wikimedia.org/wiki/File:Overfitting_svg.svg

Solutions for overfitting

- More training data
- Reduce the number of features
- Regularization
- **Dropout !**

Dropout



(a) Standard Neural Net

(b) After applying dropout.

http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf

Code: mnist_nn_dropout

```python
...

# nn layers
linear1 = torch.nn.Linear(784, 512, bias=True)
linear2 = torch.nn.Linear(512, 512, bias=True)
linear3 = torch.nn.Linear(512, 512, bias=True)
linear4 = torch.nn.Linear(512, 512, bias=True)
linear5 = torch.nn.Linear(512, 10, bias=True)
relu = torch.nn.ReLU()
dropout = torch.nn.Dropout(p=drop_prob)

# model
model = torch.nn.Sequential(linear1, relu, dropout,
                            linear2, relu, dropout,
                            linear3, relu, dropout,
                            linear4, relu, dropout,
                            linear5).to(device)

...
```

```
Epoch: 0001 cost = 0.309925616
Epoch: 0002 cost = 0.143516496
Epoch: 0003 cost = 0.113396436
Epoch: 0004 cost = 0.092770174
Epoch: 0005 cost = 0.081650071
Epoch: 0006 cost = 0.073365353
Epoch: 0007 cost = 0.070349611
Epoch: 0008 cost = 0.061270669
Epoch: 0009 cost = 0.060892191
Epoch: 0010 cost = 0.054064836
Epoch: 0011 cost = 0.051594462
Epoch: 0012 cost = 0.048855171
Epoch: 0013 cost = 0.043751985
Epoch: 0014 cost = 0.044706535
Epoch: 0015 cost = 0.044633854
Learning finished
Accuracy: 0.9771999716758728
```

Train & eval mode

```
...
total_batch = len(data_loader)
model.train()    # set the model to train mode (dropout=True)
for epoch in range(training_epochs):
...


...
# Test model and check accuracy
with torch.no_grad():
   model.eval()    # set the model to evaluation mode (dropout=False)
...
```

model.train() & model.eval()

- Sets the module in training/evaluation mode.
- This has any effect only on certain modules. See documentations of particular modules for details of their behaviors in training/evaluation mode, if they are affected, e.g. `Dropout`, `BatchNorm`, etc.

https://pytorch.org/docs/stable/nn.html?highlight=eval#torch.nn.Module.eval

What’s Next?

- Batch Normalization