# Supervised Pre-training

Sargur N. Srihari srihari@cedar.buffalo.edu

### Topics in Optimization for Deep Models

- Importance of Optimization in machine learning
- How learning differs from optimization
- Challenges in neural network optimization
- Basic Optimization Algorithms
- Parameter initialization strategies
- Algorithms with adaptive learning rates
- Approximate second-order methods
- Optimization strategies and meta-algorithms

# Topics in Optimization Strategies and Meta-Algorithms

- 1. Batch Normalization
- 2. Coordinate Descent
- 3. Polyak Averaging
- 4. Supervised Pretraining
- 5. Designing Models to Aid Optimization
- Continuation Methods and Curriculum Learning

### Topics in Pre-training

- Motivation
- The pre-training method
- Greedy supervised pre-training
- Training each layer separately
- Pre-training a very deep convolutional net
- Extension to transfer learning
- FitNets: students, teachers and hints

### Motivation

- Sometimes, directly training a model to solve a specific task can be too ambitious, if:
  - Model is too complex and hard to optimize, or
  - If the task is very difficult
- It may be more effective to
  - Train a simpler model to solve the task, then move on to confront the final task
  - Methods collectively known as pretraining

### The Pretraining Method

- Strategies that involve training models on simple tasks
- Before confronting the challenge of training the desired model to perform the desired task
- Known collectively as pretraining

### **Greedy Supervised Pretraining**

- Greedy Algorithm:
  - 1. Break a problem into many components
  - 2. Solve for the optimal version of each component in isolation
  - 3. Combine the solutions

### Role of a Greedy Algorithm

- Combining the component solutions may not yield an optimal complete solution
- However, greedy algorithms can be computationally much cheaper than algorithms that solve for the best joint solution
- Quality of a greedy solution is often acceptable if not optimal
- Initializing the joint optimization algorithm with a greedy solution can speed it up and improve the quality of the solution

### **Greedy Supervised Pretraining**

- Pretraining, is ubiquitous in deep learning.
- Pretraining algorithms that break supervised learning problems into other simpler supervised learning problems are known as greedy supervised pretraining

### Training each layer separately

- Supervised learning involving only a subset of the layers in the final neural network
- An example of greedy supervised pretraining is illustrated next
  - In which each added hidden layer is pretrained as part of a shallow supervised MLP
  - taking as input the output of the previously trained hidden layer

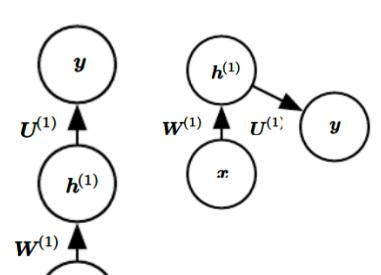
### Example of Greedy Supervised Pretraining

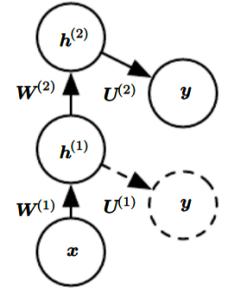
- Each hidden layer trained as a supervised MLP
  - Taking as input, output of trained hidden layer

Train a shallow architecture

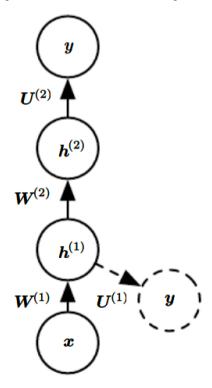
Another view of the architecture

Discard the Hidden-to-Output layer





Another drawing viewing it as a feedforward network. Then jointly fine-tune all layers



# Pretraining Deep Convolutional Nets

#### VGG pre-trained model written in Caffe

Depth increases from left (A) to right (E), as more layers are added.

Layer parameters denoted as conv( receptive field size) –(no. of channels)

ReLU activation function not shown.

ConvNet Configuration										
Α	A-LRN	В С		D	E					
11 weight	11 weight	13 weight	16 weight	16 weight	19 weight					
layers	layers	layers	layers	layers	layers					
input (224 × 224 RGB image)										
conv3-64	conv3-64	conv3-64	conv3-64	conv3-64	conv3-64					
	LRN	conv3-64	conv3-64	conv3-64	conv3-64					
maxpool										
conv3-128	conv3-128	conv3-128	conv3-128	conv3-128	conv3-128					
		conv3-128	conv3-128	conv3-128	conv3-128					
maxpool										
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256					
conv3-256	conv3-256	conv3-256	conv3-256	conv3-256	conv3-256					
			conv1-256	conv3-256	conv3-256					
conv										
	maxpool									
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512					
			conv1-512	conv3-512	conv3-512					
					conv3-512					
			pool							
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512					
conv3-512	conv3-512	conv3-512	conv3-512	conv3-512	conv3-512					
			conv1-512	conv3-512	conv3-512					
					conv3-512					
			pool							
FC-4096										
FC-4096										
FC-1000										
soft-max										

https://arxiv.org/pdf/1409.1556.pdf

Table 2: **Number of parameters** (in millions).

		,			
Network	A,A-LRN	В	C	D	E
Number of parameters	133	133	134	138	144

Instead of pre-training layer at a time, pre-train deep convolutional Network (11 weight layers)

Then use the first four and last three layers to initialize even deeper nets (with up to 19 layers of weights)

Middle layers of the new, very deep network are initialized randomly

New network is then jointly trained

# Why does greedy pre-training help?

- It helps to provide better guidance to the intermediate levels of a deep hierarchy
- Pre-training may help both in terms of optimization and in terms of generalization

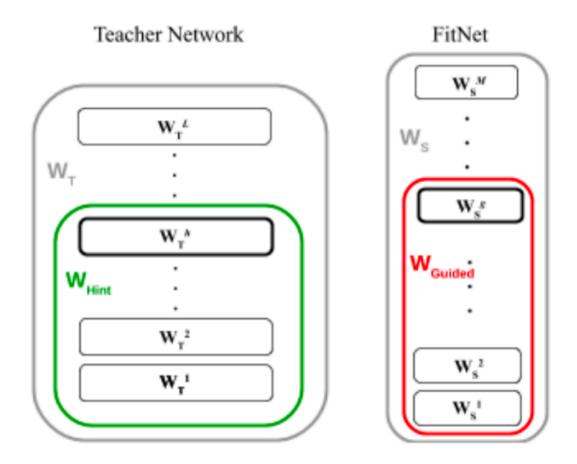
### Extension to Transfer Learning

- Pretraining extends the idea to transfer learning
- Pretrain convolutional net with 8 layers on tasks
  - (subset of 1000 ImageNet object categories)
- Then initialize same-size network with the first k
  layers of the first net
  - All layers of second network (with upper layers initialized randomly) are then jointly trained to perform a different set of tasks
    - (another subset of 1000 ImageNet categories), with fewer training examples than for the first set of tasks

### **FitNets**

- While depth improves performance, it also makes gradient-based training more difficult since deeper networks are more non-linear.
- Solution is to train a network with low enough depth and great enough width (no. of units per layer) to be easy to train
- This network becomes a teacher for a second network, designated the student
  - Student network is much deeper and thinner (11-19 layers) and would be difficult to train with SGD

### **Teacher and Student Networks**

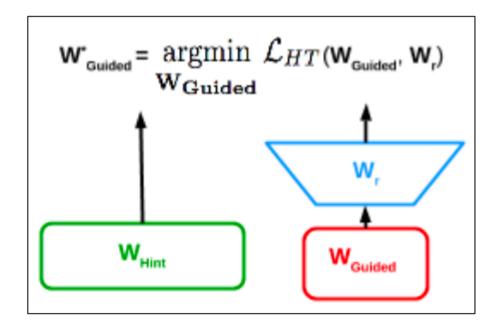


### Training the student network

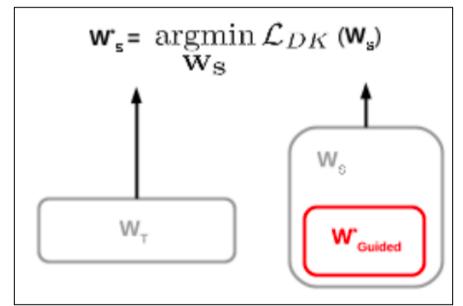
- Task is made easier by training student network not only to predict output for original task, but also to predict value of middle layer of the teacher network
- This extra task provides a set of hints about how the hidden layers should be used and can simplify the optimization problem
- Additional parameters are introduced to regress the middle layer of the 5-layer teacher network from the middle layer of the deeper student network

# Hints Training

#### Hints Training



#### **Knowledge Distillation**



### Predicting Intermediate Layers

- Instead of predicting the final classification target, the objective is to predict the middle hidden layer of the teacher network
- Objectives of Lower layers of student network:
  - 1. Help outputs of student network accomplish task
  - 2. Predict intermediate layer of the teacher network
- Although a thin-deep network may be more difficult to train than a wide-shallow network,
  - former may generalize better and has lower computational cost if it is thin enough to have far fewer parameters.

### Importance of Hints

- Without the hints on the hidden layer, the student network performs very poorly in the experiments, both on the training and test set
- Hints on middle layers may thus be one of the tools to help train neural networks that otherwise seem difficult to train, but other optimization techniques or changes in the architecture may also solve the problem.