Hopping onto the Deep Learning bandwagon

Youssef Kashef
Machine Intelligence I
08. December 2016

Outline

- What is Deep Learning?
- Neural Networks make a comeback
- Popular Network Architectures
  - Auto-encoders
  - Restricted Boltzmann machines
  - Convolutional Neural Networks
- Benefits and Challenges
- Hands-on Deep Learning
- Deep Learning at

What is Deep Learning?

The typical Machine Learning pipeline:

What is Deep Learning?

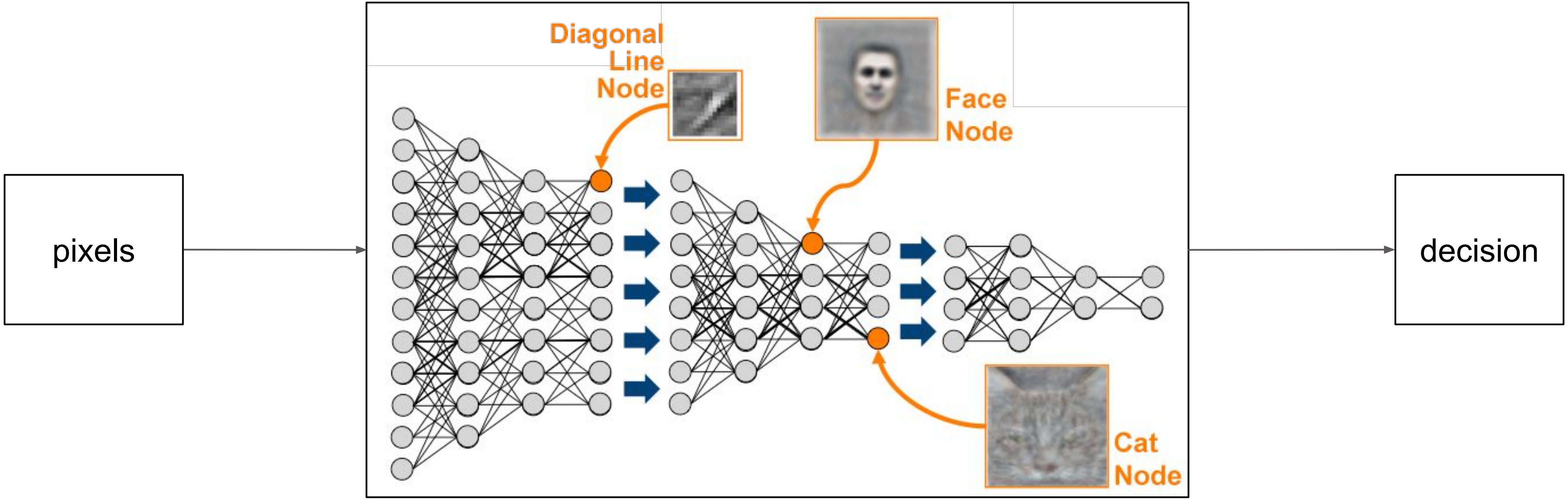
Learning a hierarchical representation of the data, directly from the data

- pixel, motif, part, object, scene, ...
- character, word, clause, sentence, ...
- audio, band, phoneme, word, phrase, ...

raw data → deep learning model → decision

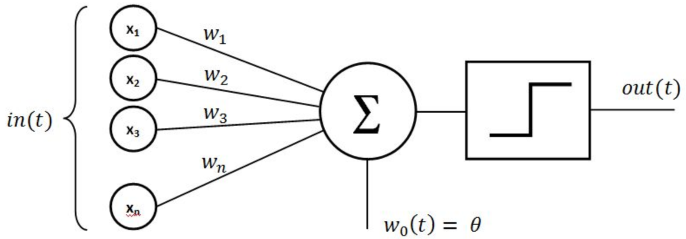
What is Deep Learning?

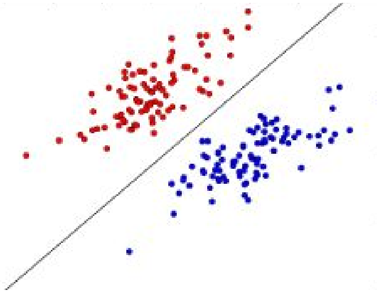
Learning a hierarchical representation of the data, directly from the data



Deep Learning using Neural Nets

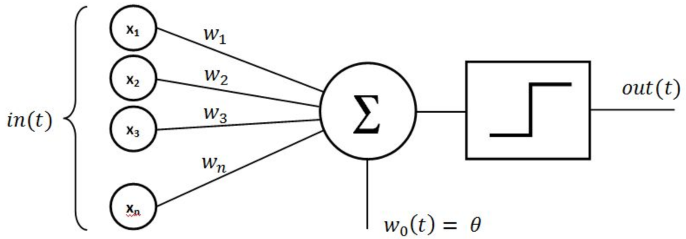
Good-old fashioned perceptron

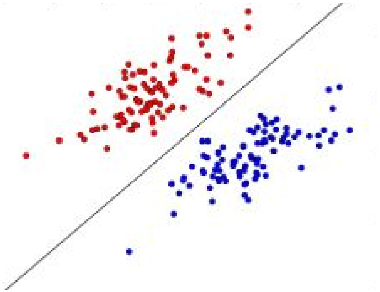




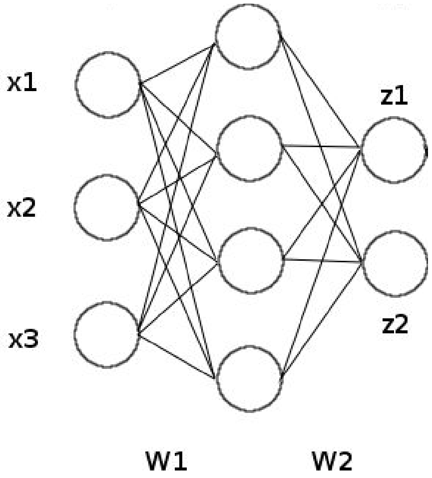
Deep Learning using Neural Nets

Good-old fashioned perceptron



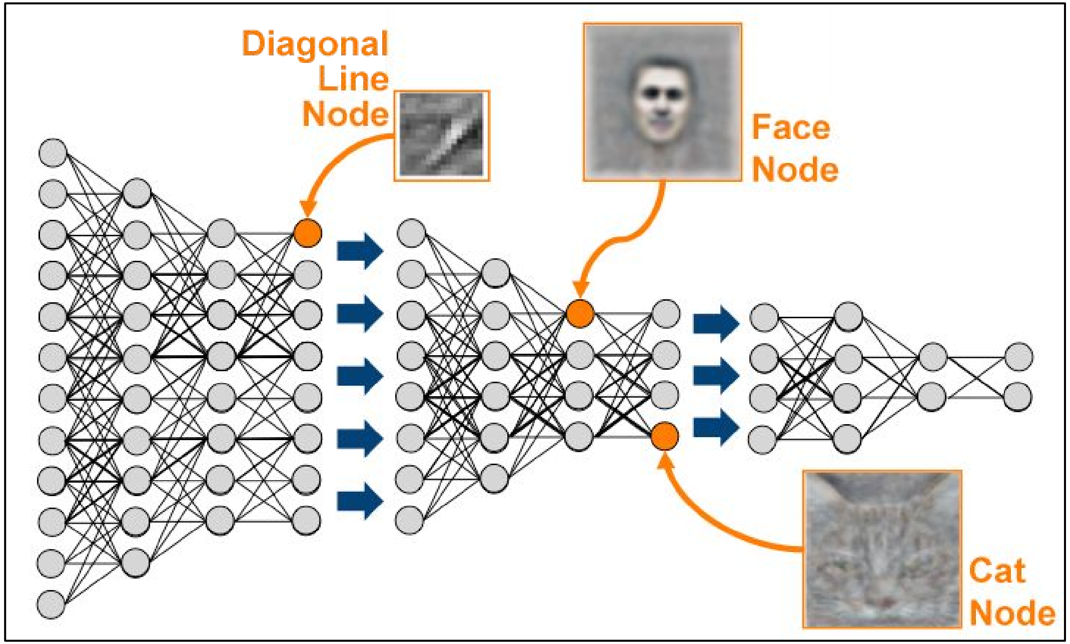


Good-old MLP



Deep Learning using Neural Nets

MLP with many many layers → Deep Neural Net



Deep Learning using Neural Nets

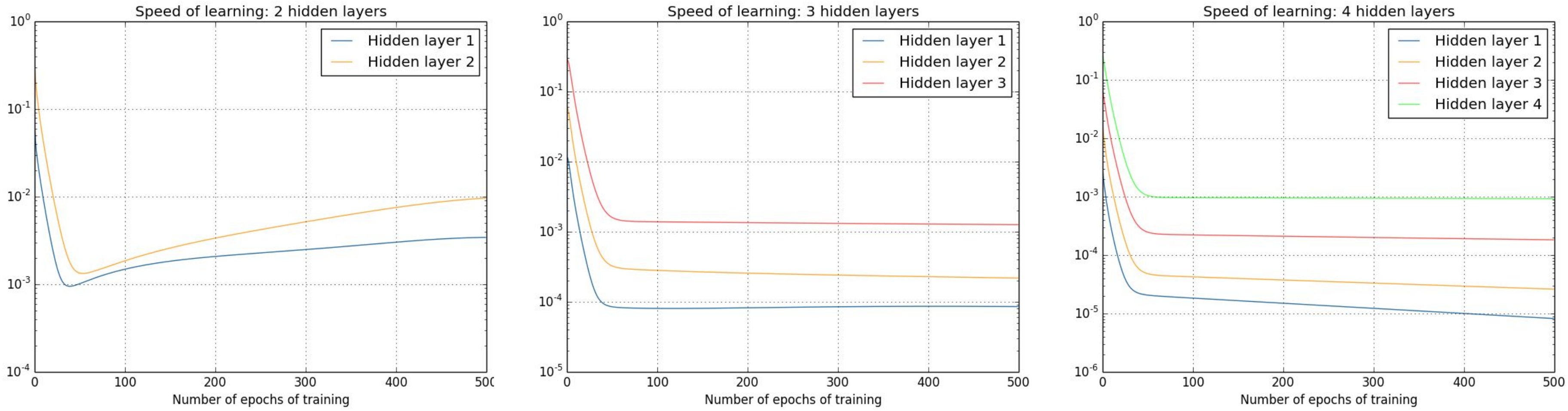
MLP with many many layers → Deep Neural Net

But when training it the usual way,
not any better than 2-layer MLP

- Some weights get stuck
- Some weights are unstable
- gradients in one layer are very different from the next

Deep Learning using Neural Nets

The Vanishing Gradient Problem



[Nielsen15]

Deep Learning using Neural Nets

MLP with many many layers → Deep Neural Net

But:

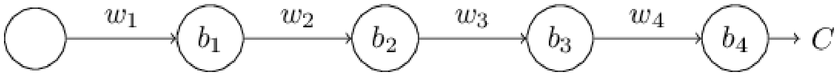
- Vanishing/exploding of gradients
- Random initializations limiting
- Overfitting

Deep Learning using Neural Nets

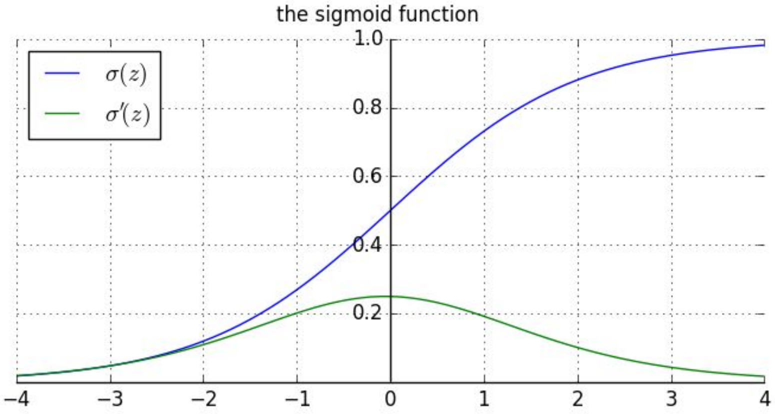
MLP with many many layers → Deep Neural Net

But:

- Vanishing/exploding of gradients
- Random initializations limiting
- Overfitting



$$\frac{\partial C}{\partial b_1} = \sigma'(z_1)\, w_2 \sigma'(z_2)\, w_3 \sigma'(z_3)\, w_4 \sigma'(z_4)\, \frac{\partial C}{\partial a_4}$$

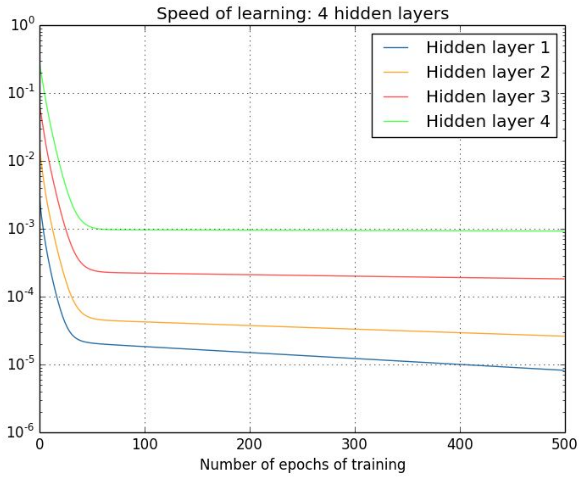


Deep Learning using Neural Nets

MLP with many many layers → Deep Neural Net

But:

- Vanishing/exploding of gradients
- Random initializations limiting
- Overfitting

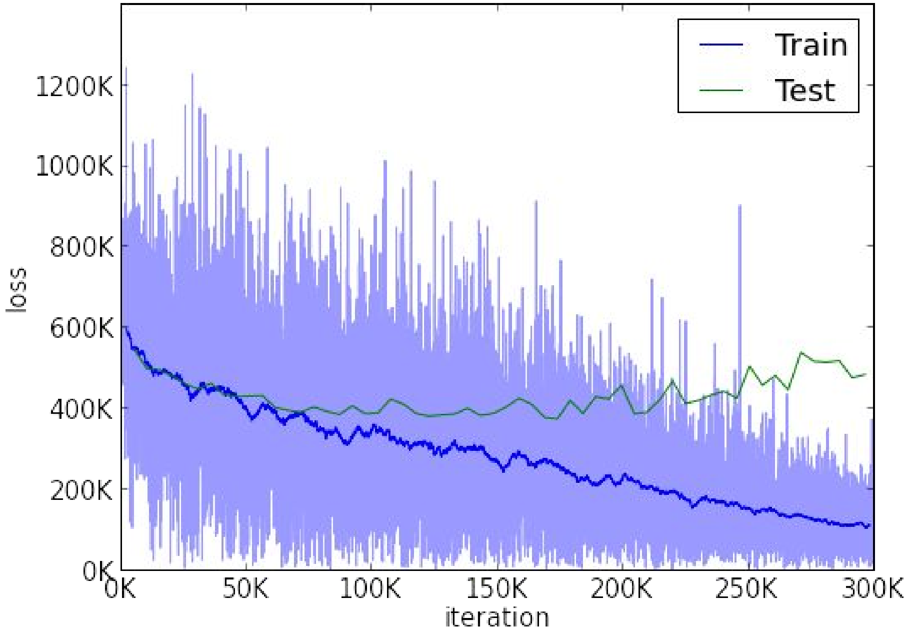


Deep Learning using Neural Nets

MLP with many many layers → Deep Neural Net

But:

- Vanishing/exploding of gradients
- Random initializations limiting
- Overfitting



Deep Learning using Neural Nets

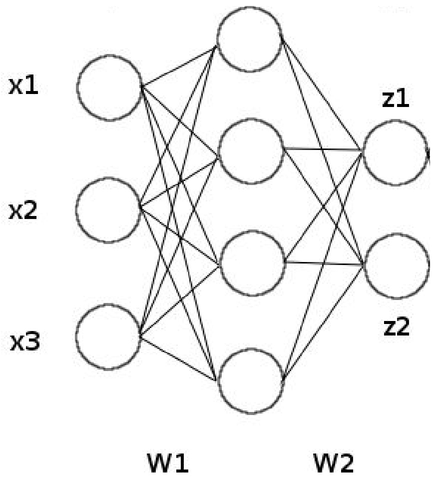
MLP with many many layers → Deep Neural Net

But:

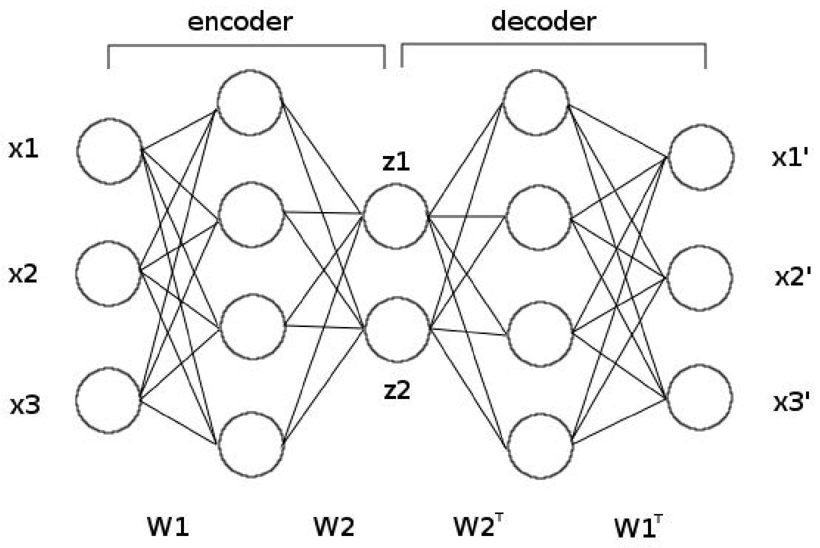
- Vanishing/exploding of gradients ✓ non-saturating activation function (e.g. ReLU)
- Random initializations limiting ✓ unsupervised pre-training (more in just a few)
- Overfitting ✓ data, **LOTS** of data, but not completely solved

Popular Architectures: Autoencoders

Good-old MLP

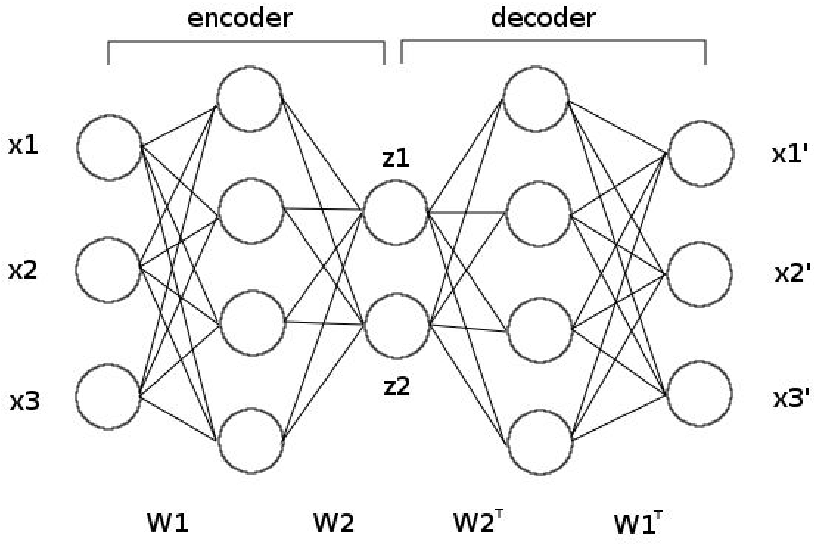


Popular Architectures: Autoencoders



$$h_{W,b}(x) \approx x$$

Popular Architectures: Autoencoders



$$h_{W,b}(x) \approx x$$

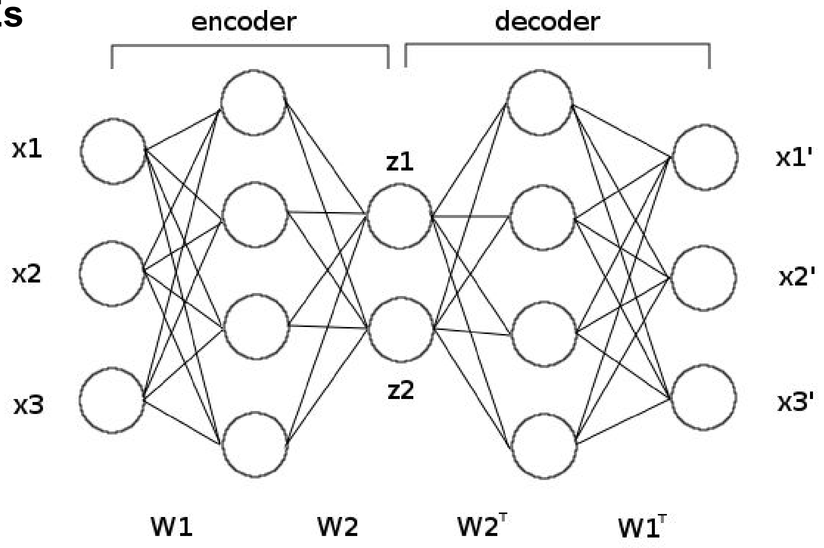
learn representations, NOT identity

How?

Popular Architectures: Autoencoders

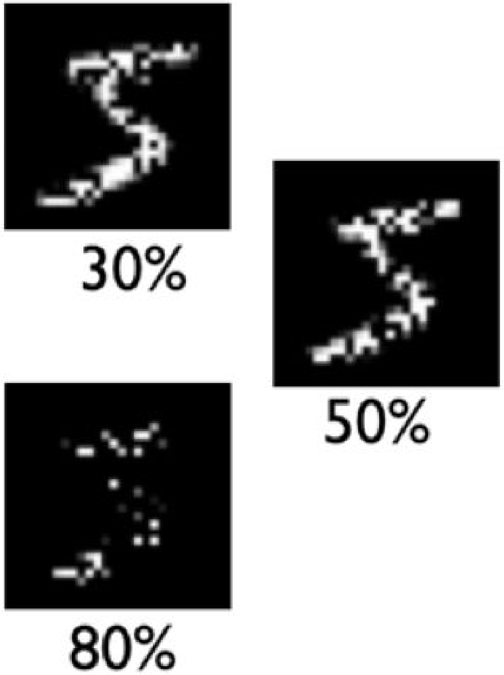
How to avoid learning the identity function?

- **Undercomplete AEs**
- **Regularization**
- **Denoising**

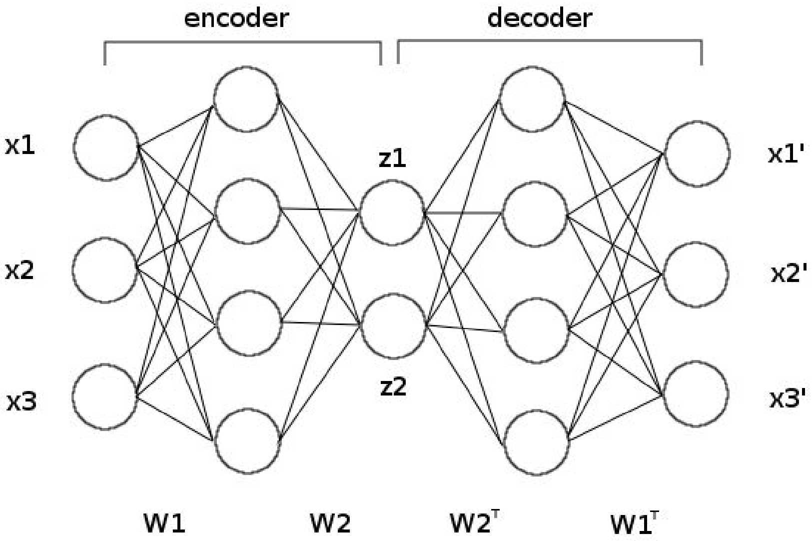


$$h_{W,b}(x) \approx x$$

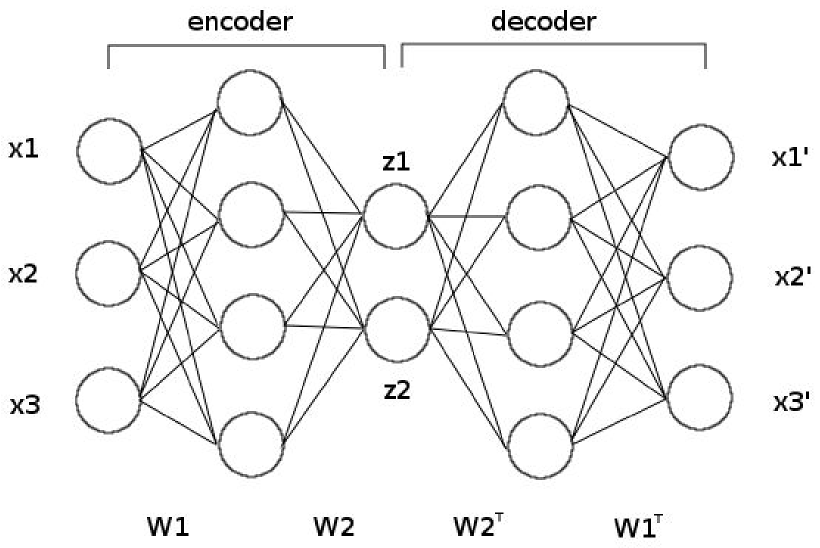
Autoencoders → Denoising Autoencoders



Reconstruct variables from corrupted input

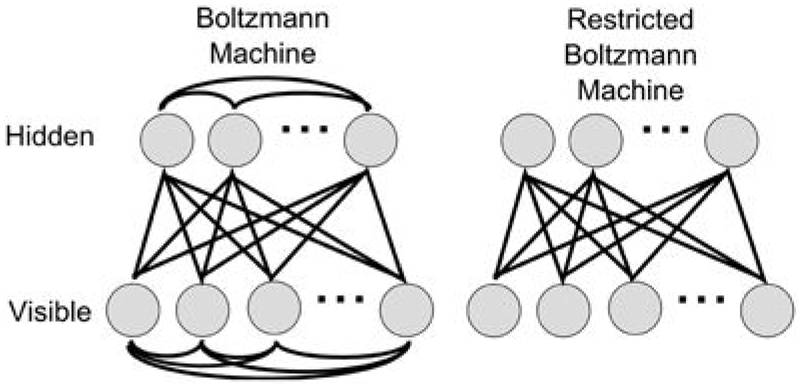


Autoencoders → Stacked Denoising Autoencoders

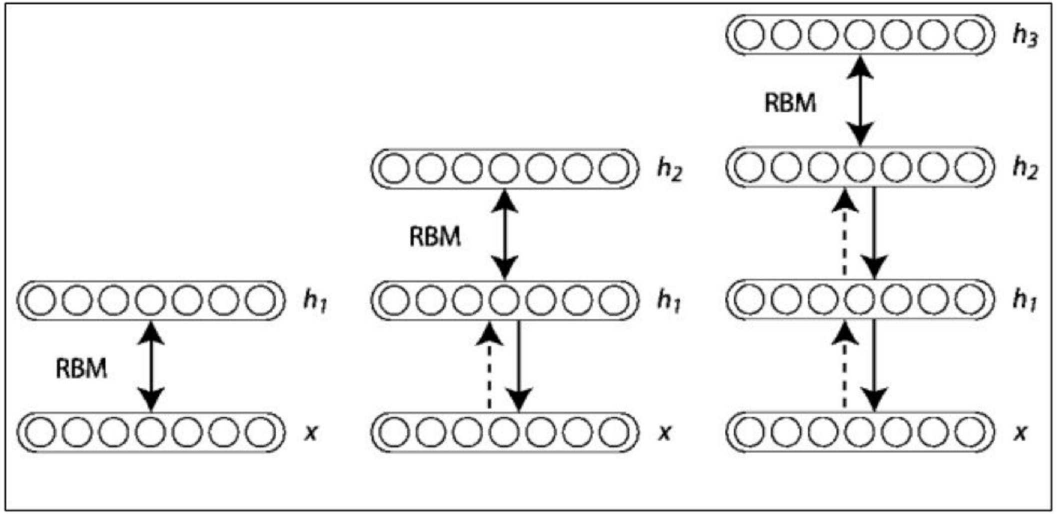


$$h_{W,b}(x) \approx x$$

Popular Architectures: Restricted Boltzmann Machines (RBM)

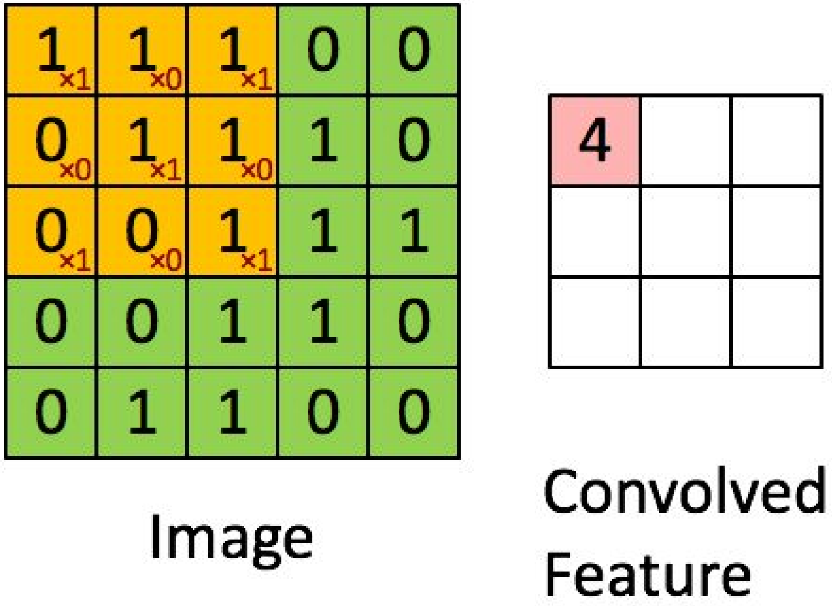


RBM → Deep Belief Nets



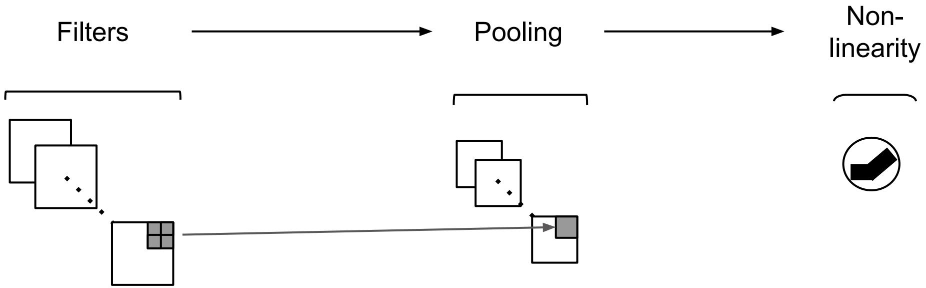
Popular Architectures: Convolutional Neural Networks

2D ~~Convolution~~ Correlation



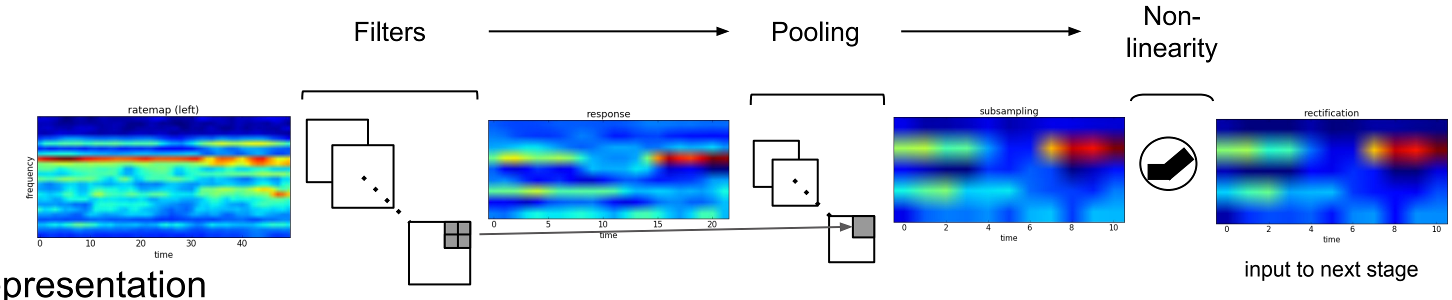
Popular Architectures: Convolutional Neural Networks

The convolutional stage:



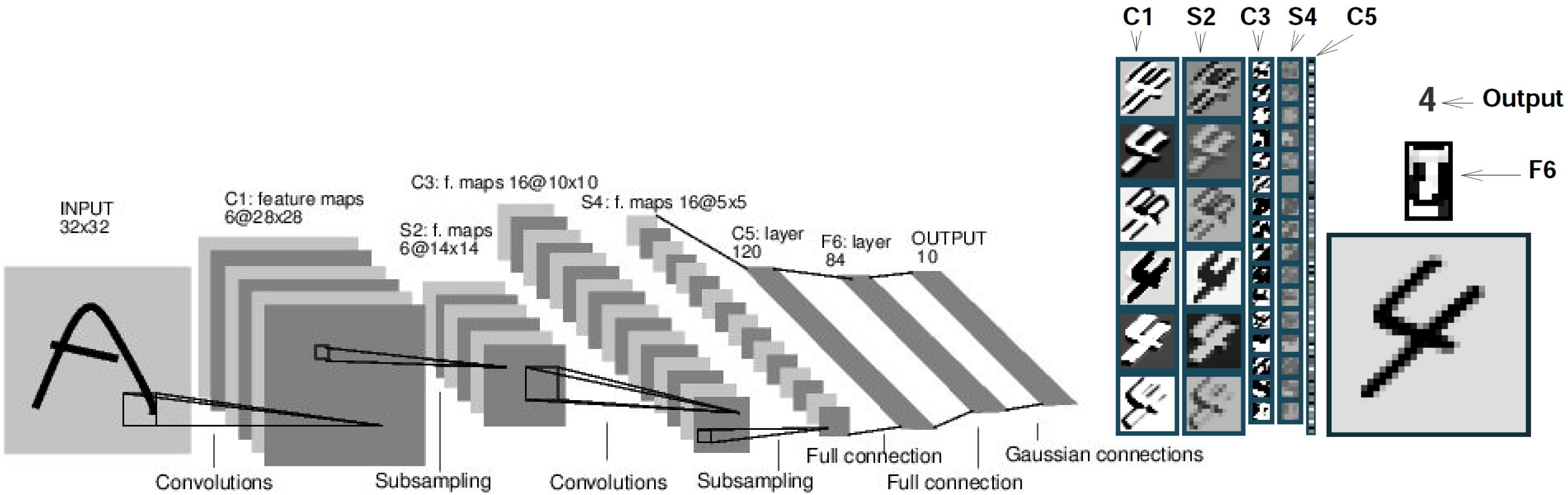
Popular Architectures: Convolutional Neural Networks

The convolutional stage:



Spatio-temporal representation of an audio signal (e.g. STFT, ratemaps, …)

Convolutional Neural Networks



[LeCun et al.]

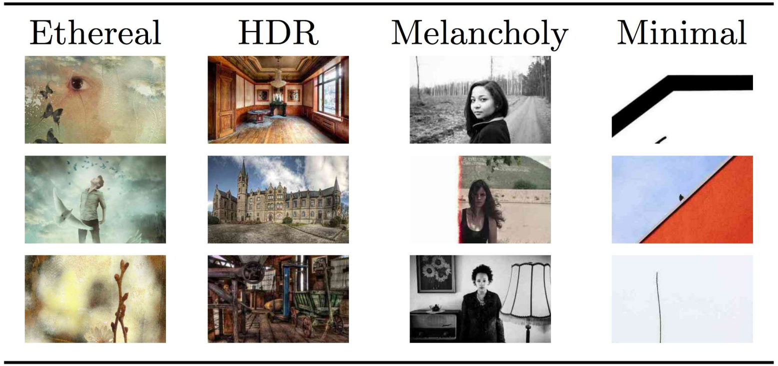
Payoffs and Challenges

Payoffs

- Implicit Feature Learning
- Same data, same model, different tasks

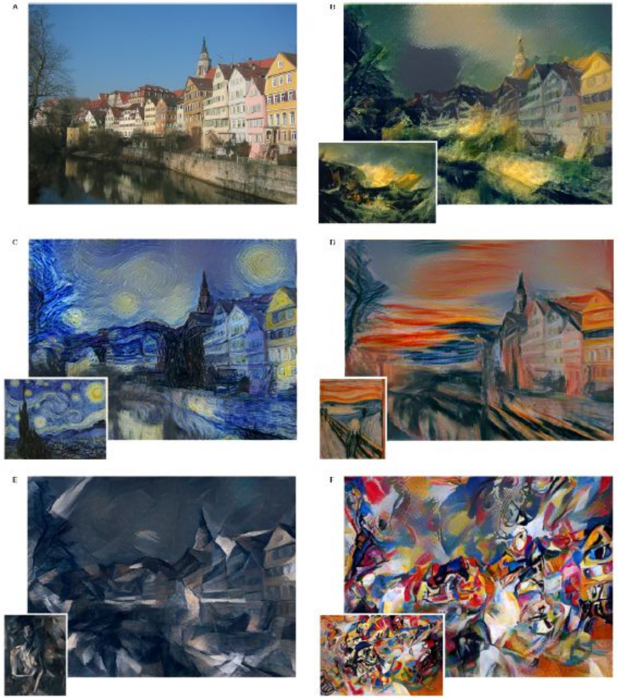
Payoffs and Challenges

Visual Style Recognition



[Karayev14]

Re-paint in style



[Gatys15]

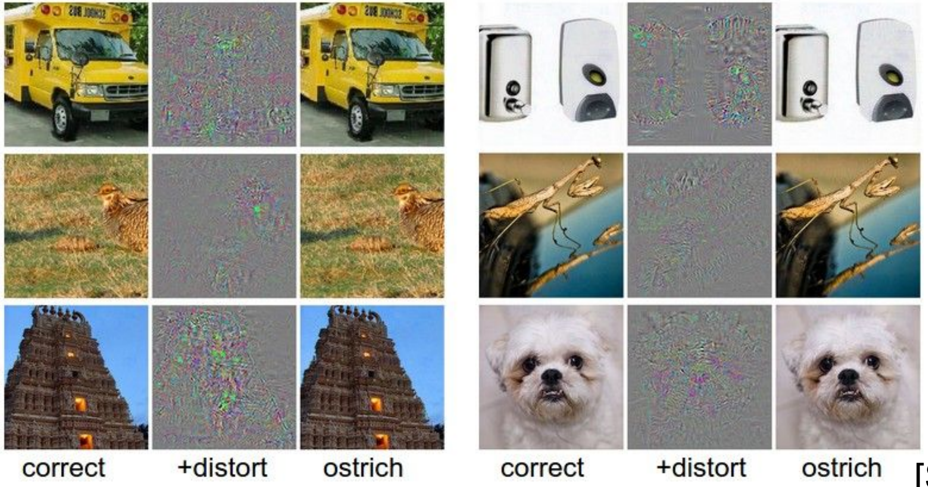
Payoffs and Challenges

Payoffs

- Implicit Feature Learning
- Same data different tasks

Challenges

- Computationally intensive
- Prone to overfitting
- adversarial images



[Szegedy et al.]

Payoffs and Challenges

Challenges

- Computationally intensive
- Prone to overfitting
- adverserials images

Solutions

?

Payoffs and Challenges

Payoffs

- Implicit Feature Learning
- Same data, same model, different tasks

Challenges

- Computationally intensive
- Prone to overfitting
- adverserials images

Solutions

GPU accelerated computation

data augmentation, Convolution, Dropout

Payoffs and Challenges

Payoffs

- Implicit Feature Learning
- Same data, same model, different tasks

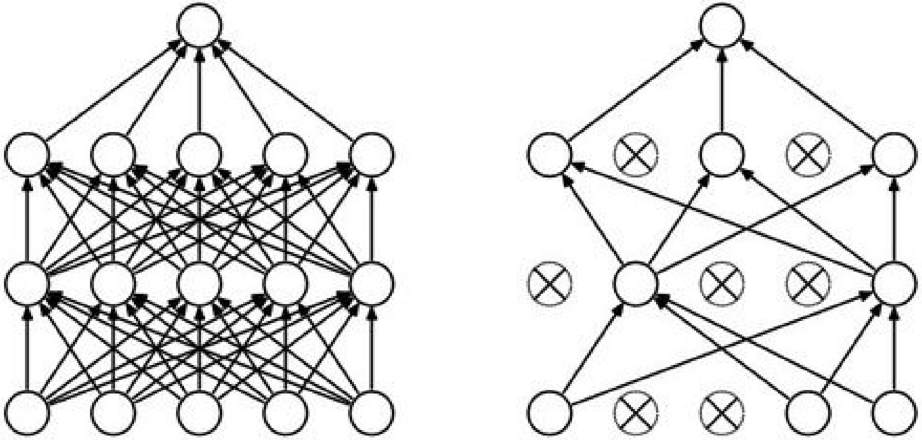
Challenges

- Computationally intensive
- Prone to overfitting
- adverserials images

Solutions

GPU accelerated computation

data augmentation, Convolution, **Dropout**



(a) Standard Neural Net (b) After applying dropout.

Hands-on Deep Learning

Open Source frameworks

dmlc
mxnet

theano

TensorFlow

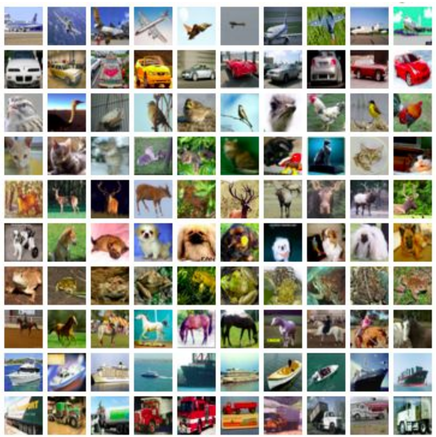
DL4J

BVLC Caffe

torch

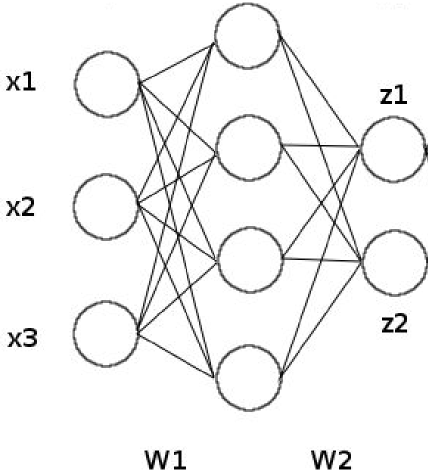
Hands-on Deep Learning using Caffe

Data



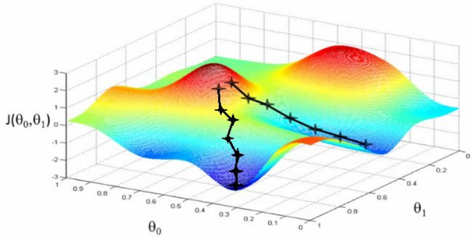
shuffle, augment, format

Model



Directed Acyclic Graph

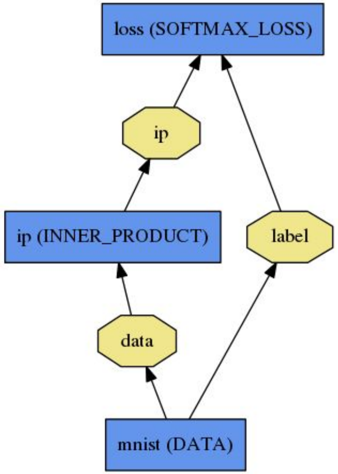
Solver



Learning Algorithm
(e.g. Gradient Descent)

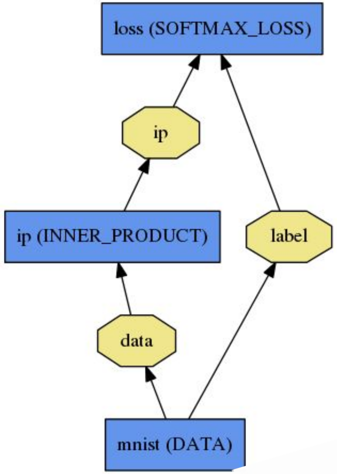
Hands-on Deep Learning using Caffe

Model

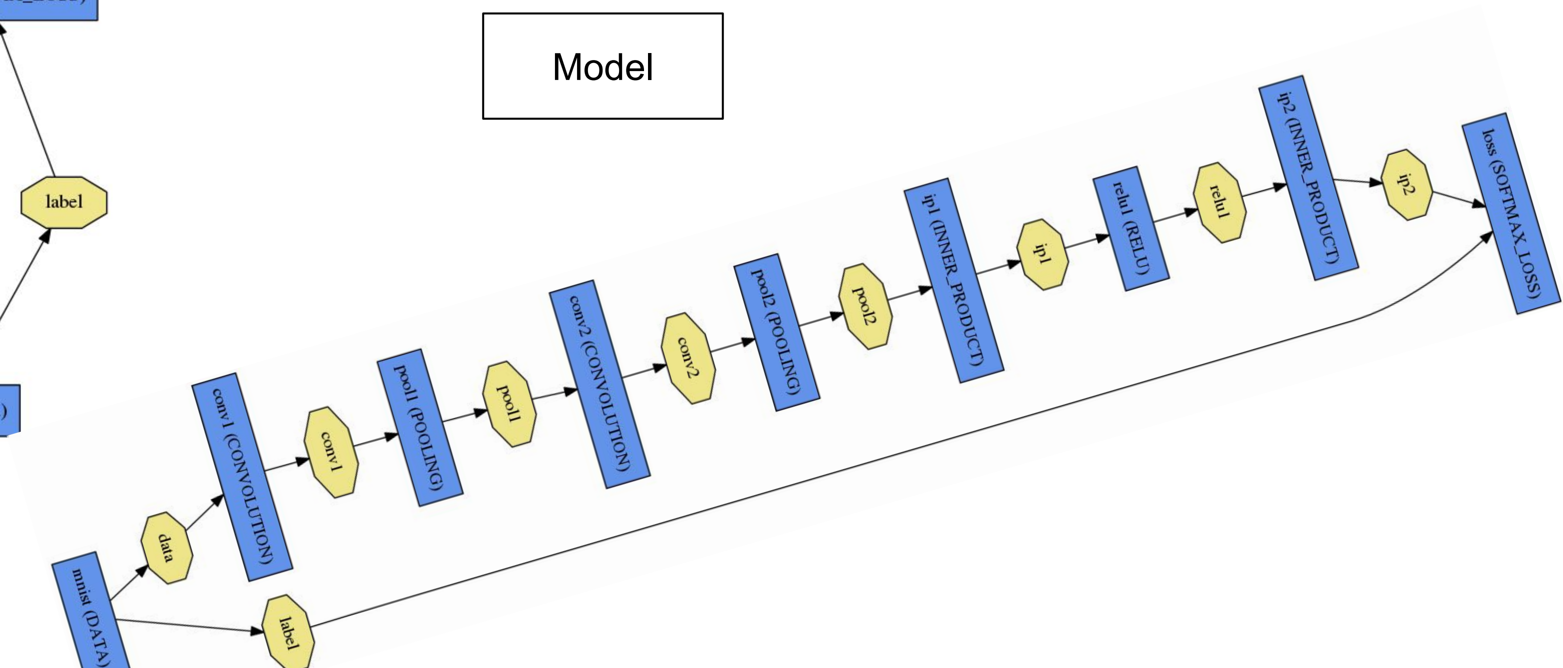


Directed Acyclic Graph

Hands-on Deep Learning using Caffe



Model

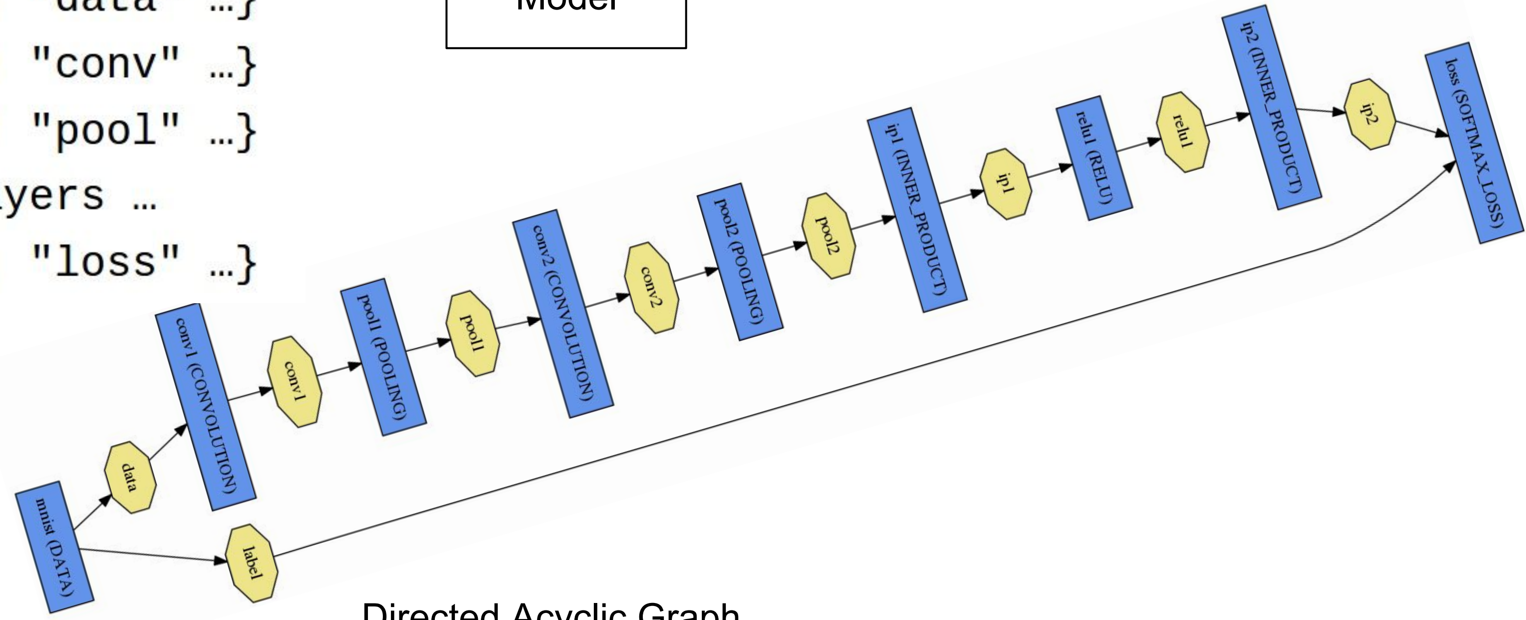


Directed Acyclic Graph

Hands-on Deep Learning using Caffe

```
name: "dummy-net"
layer { name: "data" …}
layer { name: "conv" …}
layer { name: "pool" …}
     … more layers …
layer { name: "loss" …}
```

Model



Directed Acyclic Graph

Hands-on Deep Learning using Caffe

Solver

```
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
max_iter: 10000
snapshot_prefix: "lenet_snapshot"
```

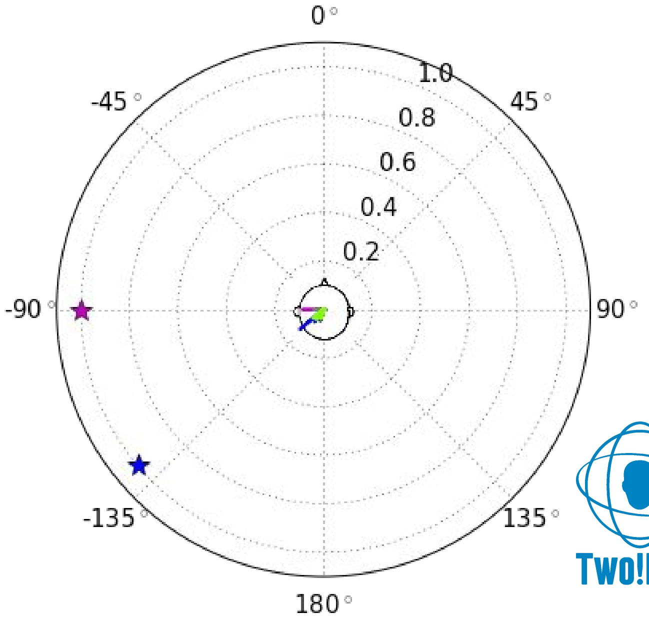
Hands-on Deep Learning using Caffe

Solver

```
I0901 13:35:20.426187 20072 solver.cpp:232] Iteration 65000, loss = 61.5498
I0901 13:35:20.426218 20072 solver.cpp:251] Iteration 65000, Testing net (#0) # train set
I0901 13:35:22.780092 20072 solver.cpp:289] Test loss: 60.8301
I0901 13:35:22.780138 20072 solver.cpp:302]     Test net output #0: cross_entropy_loss = 60.8301 (* 1 = 60.8301 loss)
I0901 13:35:22.780146 20072 solver.cpp:302]     Test net output #1: l2_error = 2.02321
```

Deep Learning at NI:
Multi-Objective Deep Learning

[Malik]



TWO!EARS

Deep Learning at NI: Multi-Objective Deep Learning

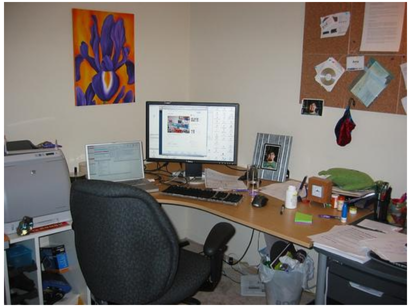
Training specialized networks from random initializations.

Re-purposing networks is possible and successful.

How do we learn a general purpose representation?

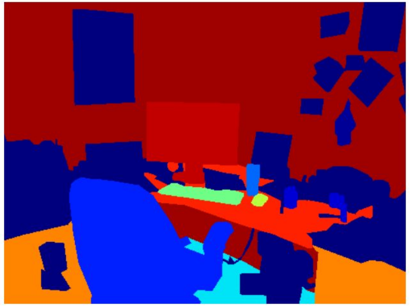
Regularization in the objective space.

Semantic Segmentation

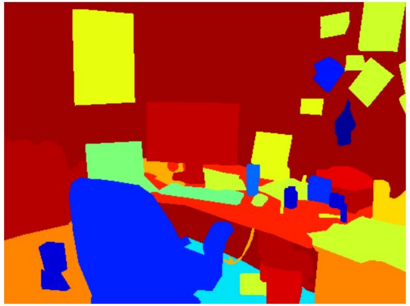


[Mottaghi]

Foreground-Background subtraction (Binary)



Stuff and **Things** (M-way)



Dense scene labeling/ scene parsing (M-way)

2

no. of classes M

large M

End-to-end learning: Fully Conv. Networks for Semantic Segmentation (CVPR 2015)

CVPR 2015

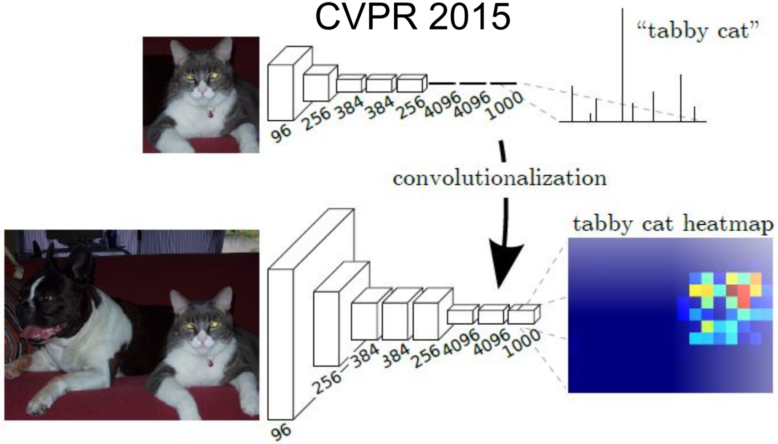
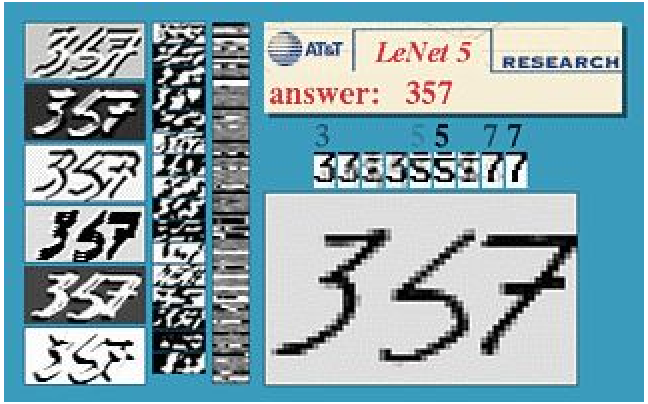


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

[Long et al.]

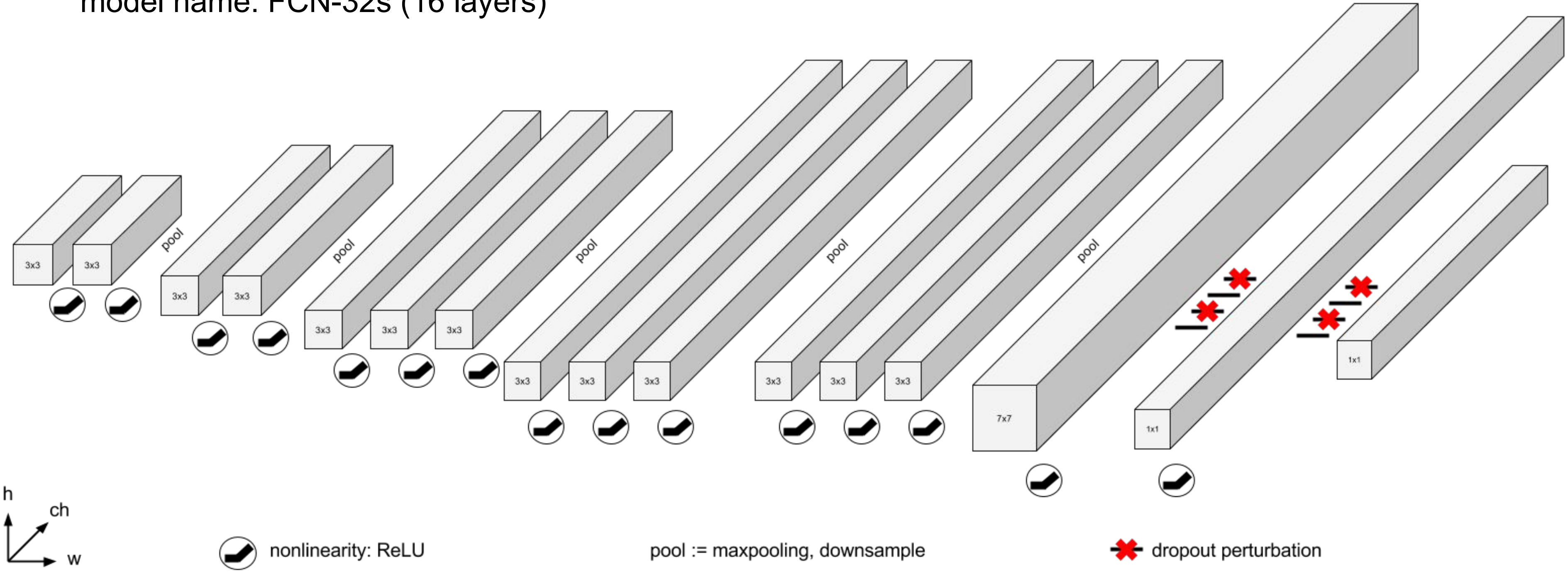
NIPS 1992



[Matan et al.]

Training Fully Conv. Networks for Semantic Segmentation

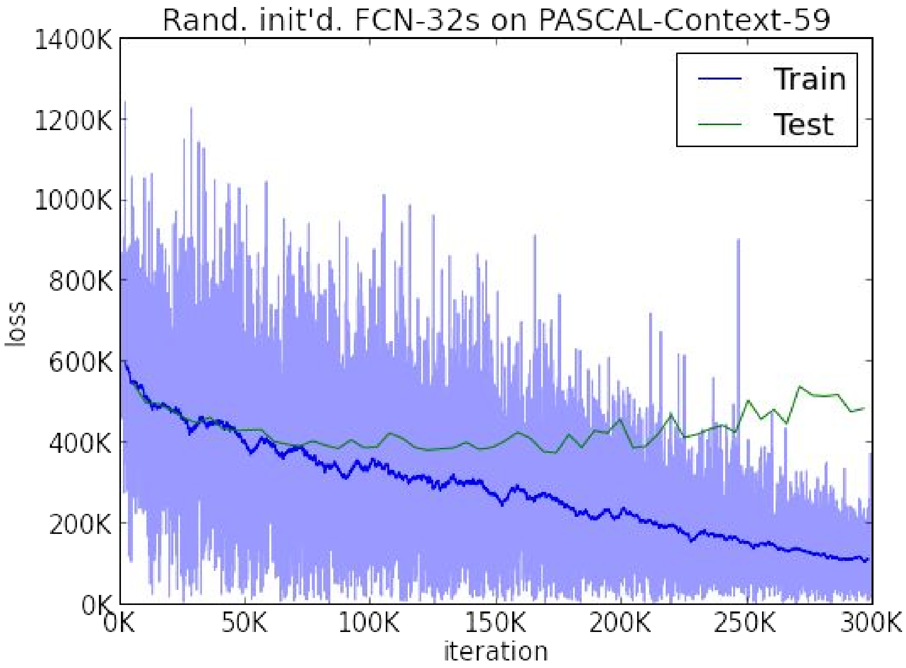
model name: FCN-32s (16 layers)

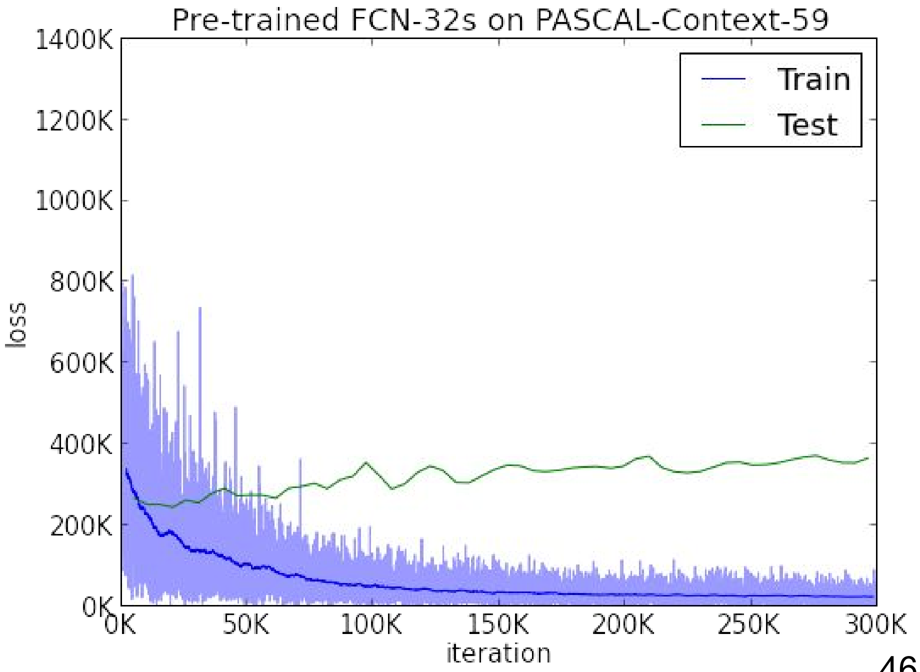


[Simonyan]

Finetuning vs. random initialization

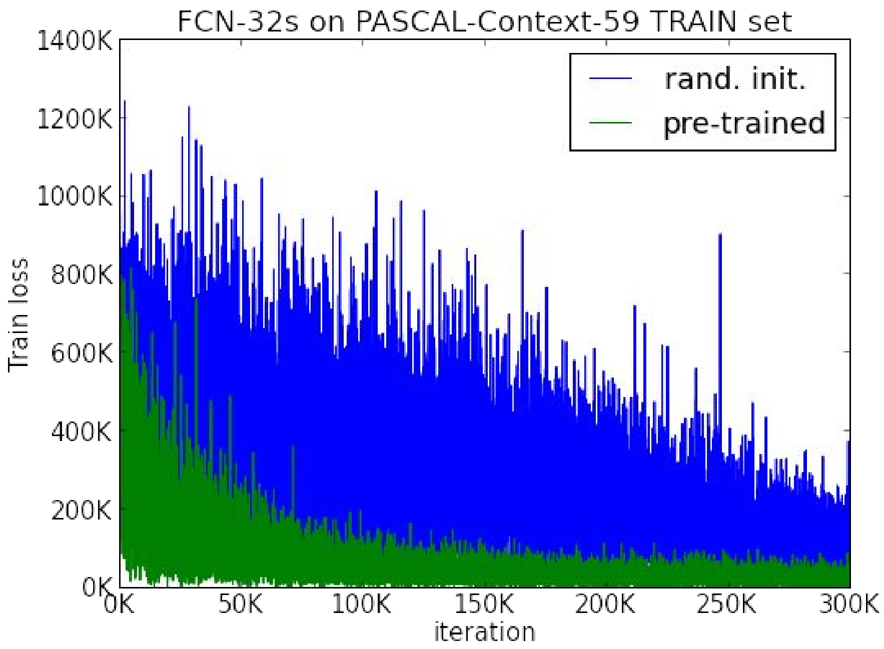
loss := multinomial logistic softmax loss (without normalization)



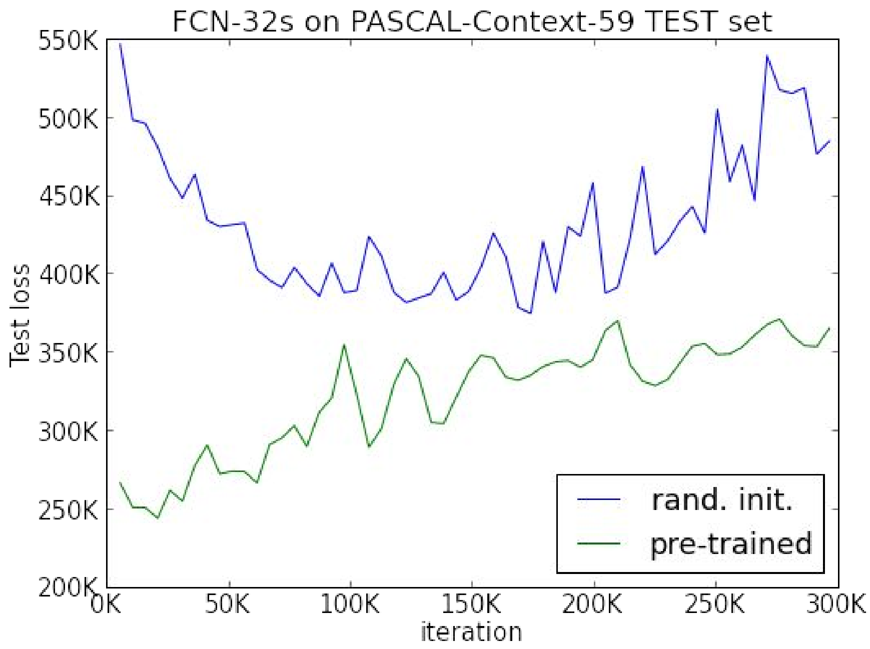


Finetuning vs. random initialization

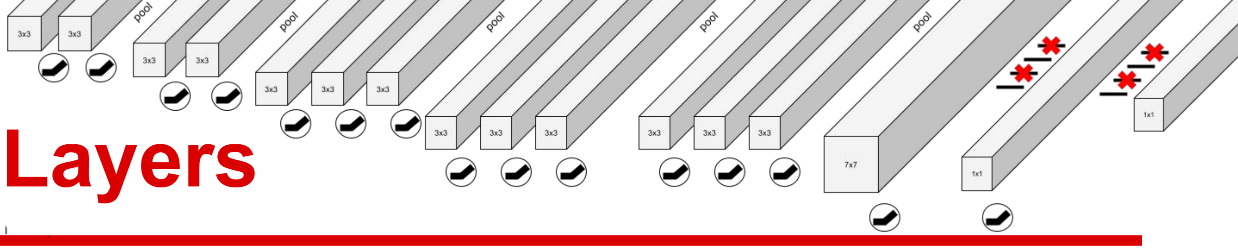
Train loss without normalization



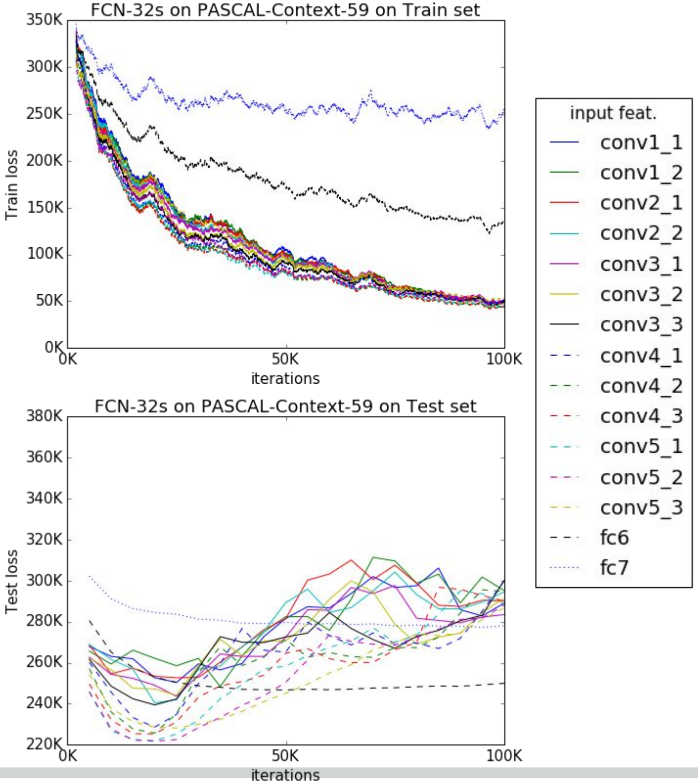
Test loss without normalization



Fixing Feature Layers

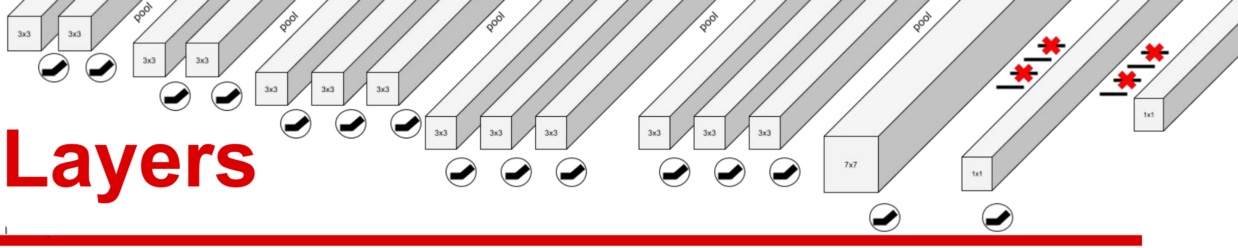


Preventing earlier weights from updating during optimization

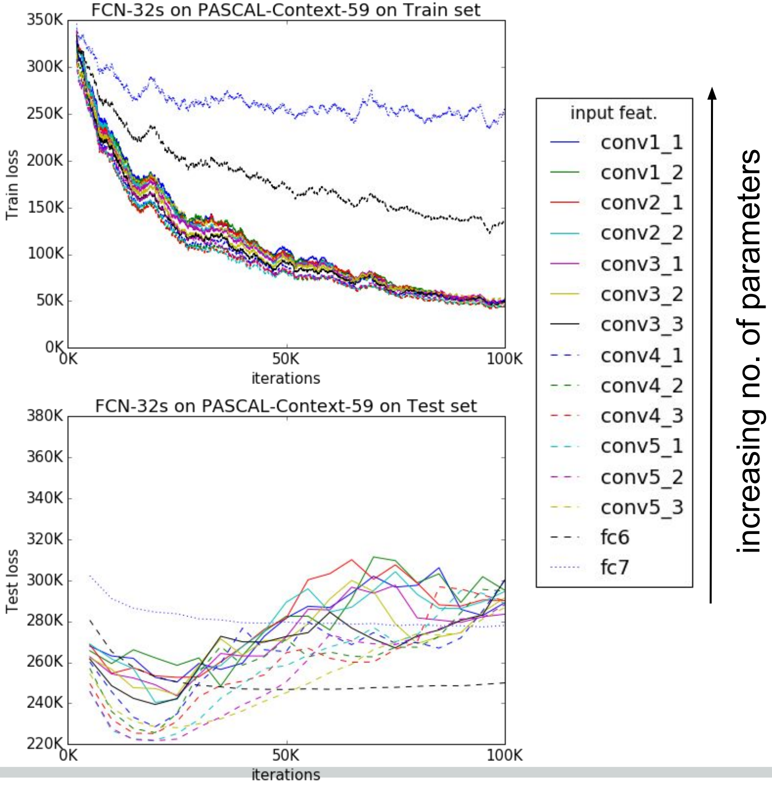


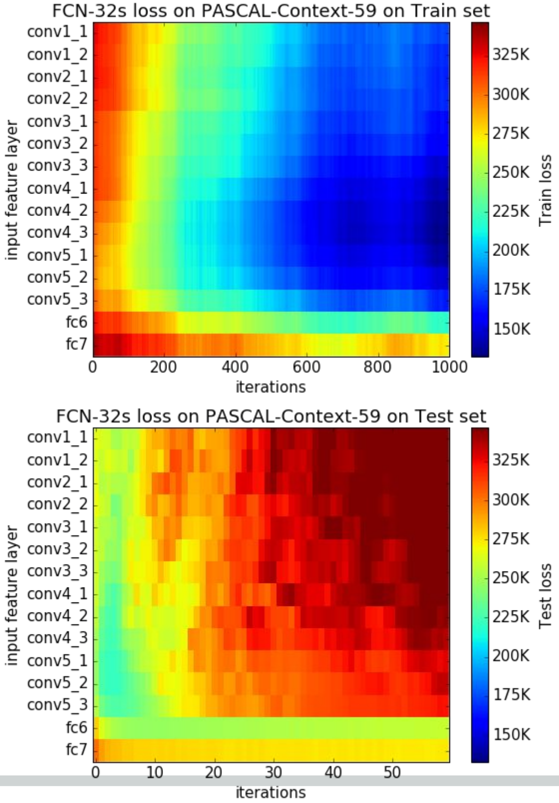
increasing no. of parameter

Fixing Feature Layers

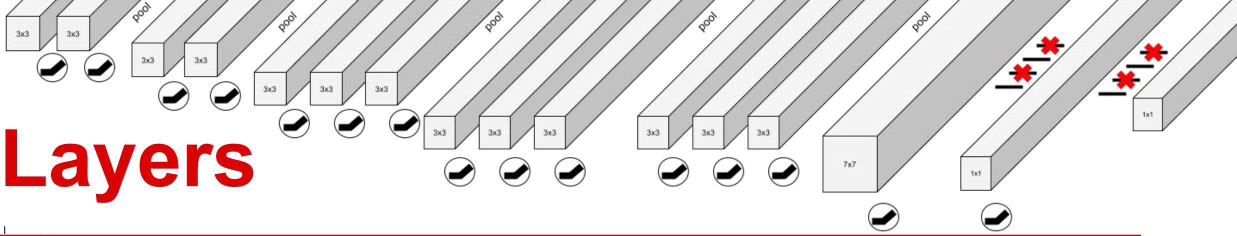


Preventing earlier weights from updating during optimization

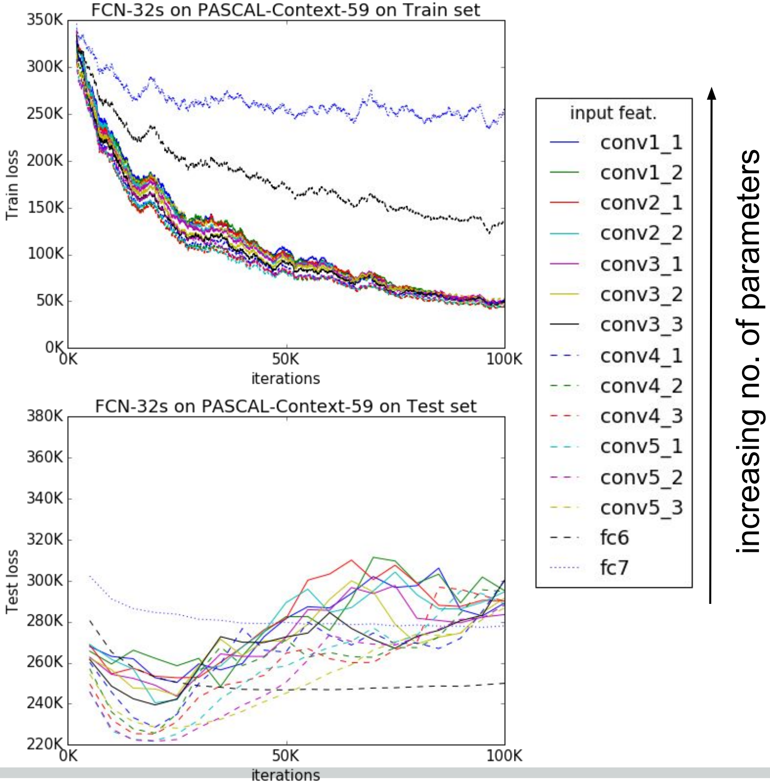


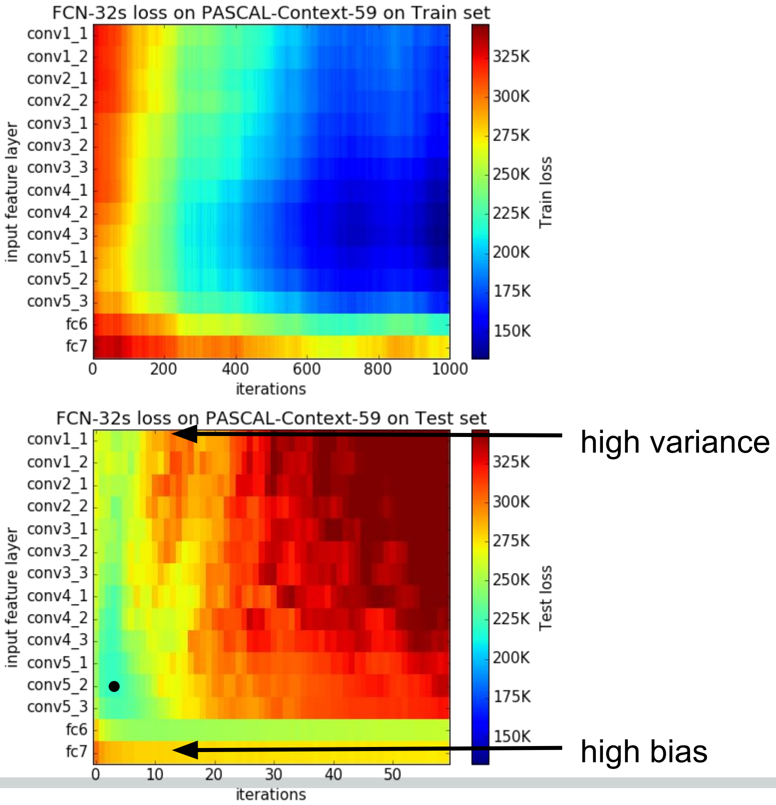


Fixing Feature Layers

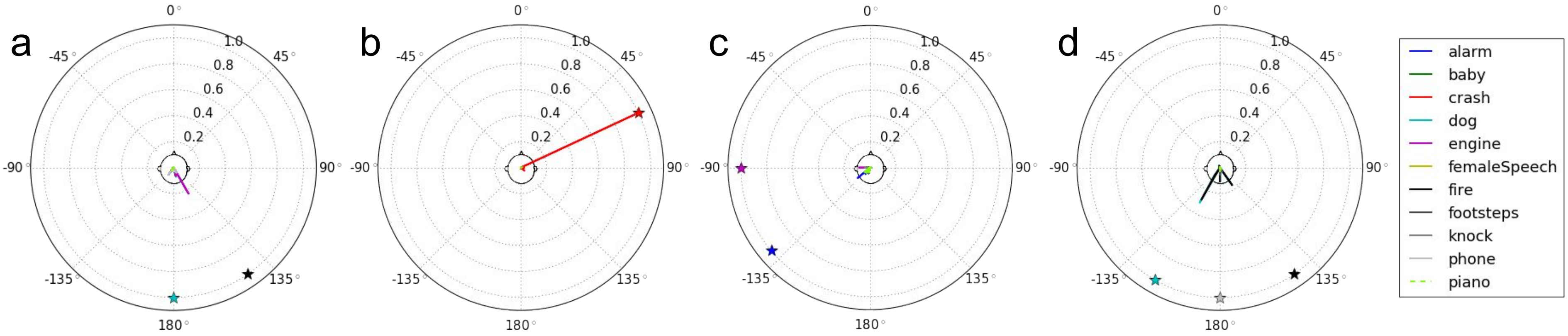


Preventing earlier weights from updating during optimization

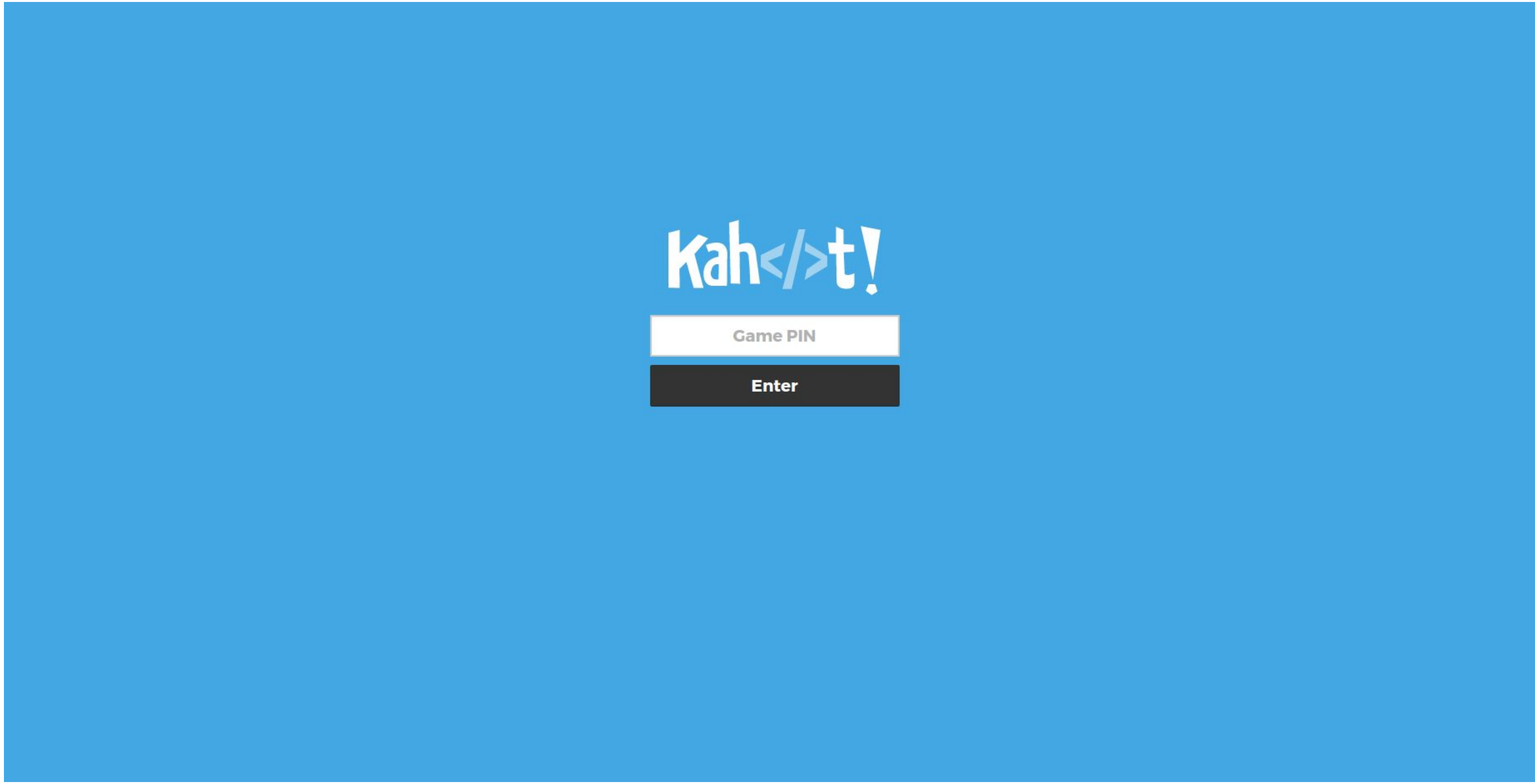




Joint sound identification and localisation



www.kahoot.it



Further Reading

Books

Goodfellow, I., Bengio Y. and Courville A. (2016). Deep Learning. Book in preparation for MIT Press.

Nielsen, M.A (2015). Neural Networks and Deep Learning. Determination Press.

Review papers

J. Schmidhuber. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks.

Tutorials and code examples

Deep Learning Tutorials. Theano.

A bit of everything

http://deeplearning.net/

References

Couprie, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor Semantic Segmentation using depth information. Iclr, 1–8. Retrieved from http://arxiv.org/pdf/1301.3572.pdf

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627–45.

Gatys, L. a, Ecker, A. S., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. arXiv, 3–7. Retrieved from http://arxiv.org/abs/1508.06576

Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., & Winnemoeller, H. (2013). Recognizing Image Style, 1–20. Retrieved from http://arxiv.org/abs/1311.3715

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. CVPR. Retrieved from http://arxiv.org/abs/1411.4038v1.

Malik, J. (2013). The Three R's of Computer Vision. UC Berkeley.

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Yuille, A. (2014). The role of context for object detection and semantic segmentation in the wild. *Cvpr*, 891–898. doi:10.1109/CVPR.2014.119

Nielsen, M.A (2015). Neural Networks and Deep Learning. Determination Press.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus. (2014). Intriguing properties of neural networks

Van Essen, D. C., & Gallant, J. L. (1994). Neural mechanisms of form and motion processing in the primate visual system. Neuron, 13(1), 1–10.