

Deeply-Supervised Nets

AISTATS, 2015

Deep Learning Workshop, NIPS 2014

Zhuowen Tu

Department of Cognitive Science

Department of Computer Science and Engineering (affiliate)

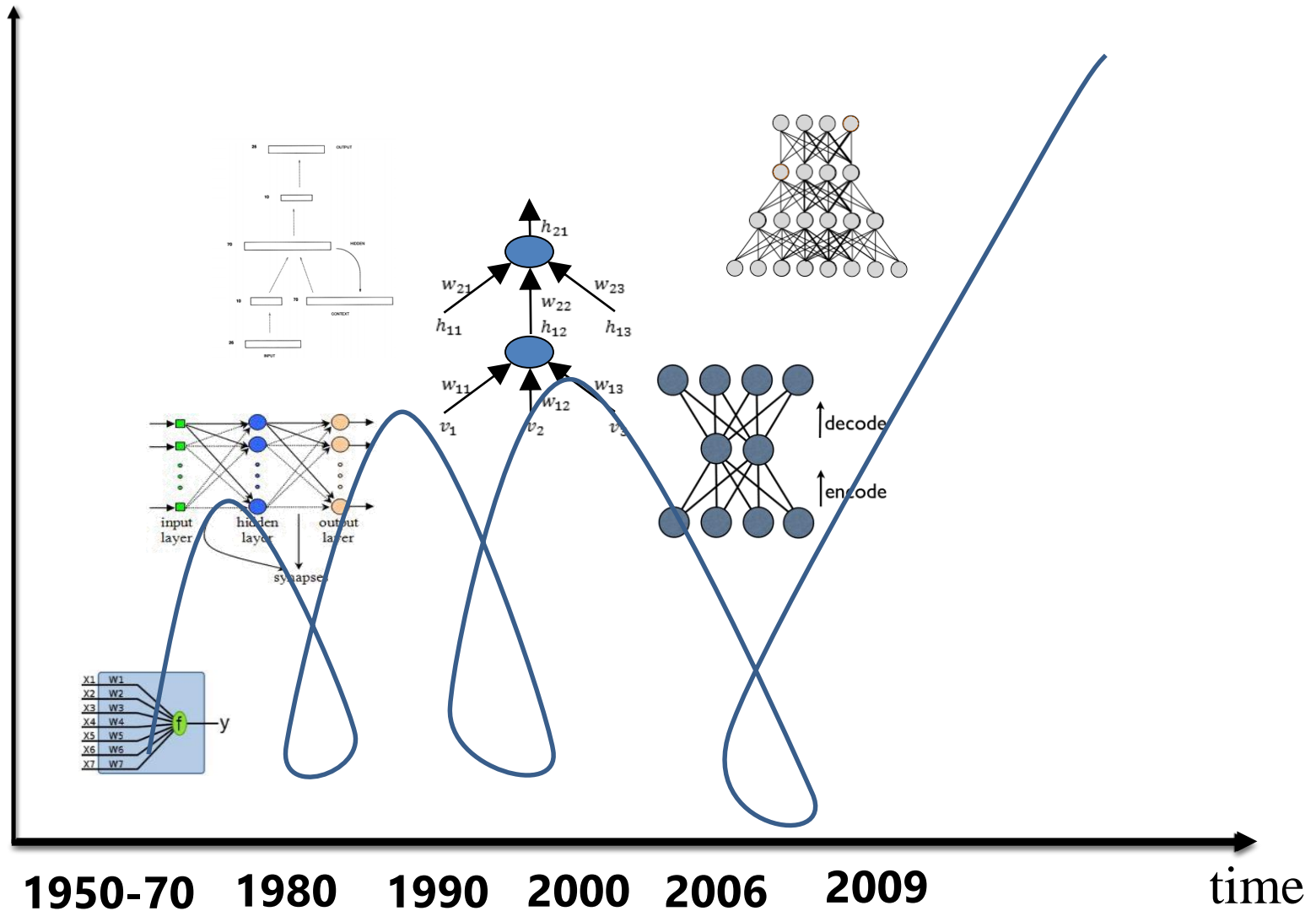
University of California, San Diego (UCSD)

with Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zengyou Zhang

Funding support: NSF IIS-1360566, NSF IIS-1360568

Artificial neural networks: a brief history

Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain".
Hopfield J. (1982), "Neural networks and physical systems with emergent collective computational abilities", PNAS.
Rumelhart D., Hinton G. E., Williams R. J. (1986), "Learning internal representations by error-propagation".
Elman, J.L. (1990). "Finding Structure in Time".
Hinton, G. E.; Osindero, S.; Teh, Y. (2006). "A fast learning algorithm for deep belief nets".



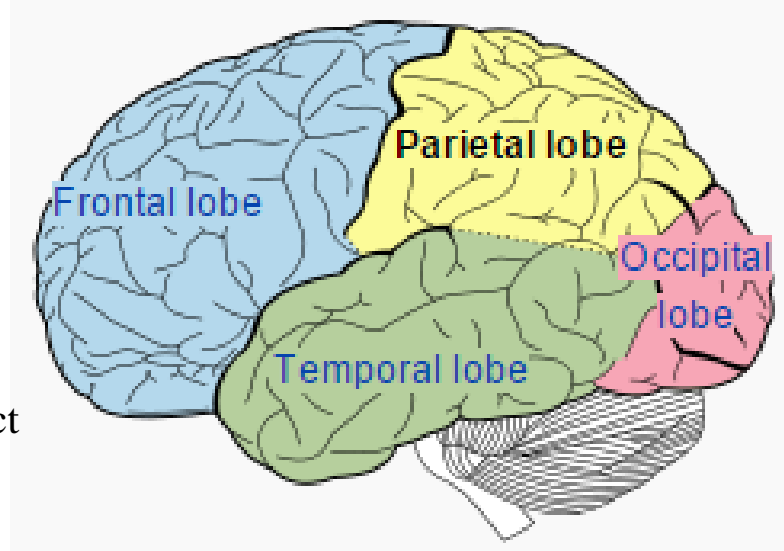
Visual representation

Frontal lobe:

- motor control,
- decisions and judgments, emotions
- language production

Temporal lobe:

- Visual perception, object recognition, auditory processing
- Memory
- Language comprehension

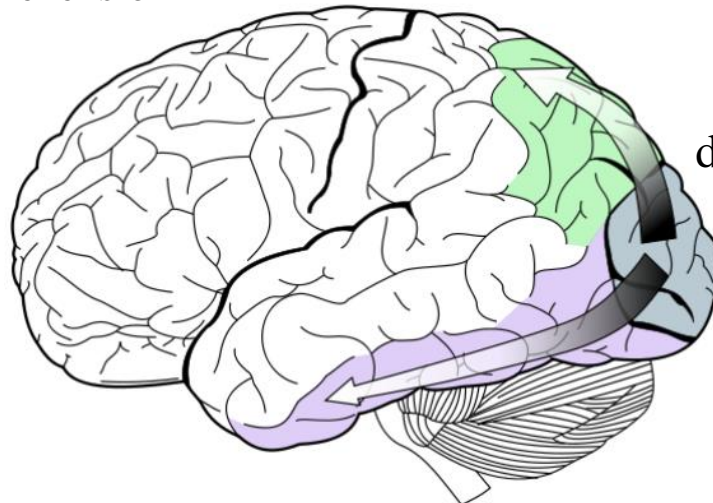


Parietal lobe:

- Attention
- Spatial cognition
- Perception of stimuli related to touch, pressure, temperature, pain

Occipital lobe:

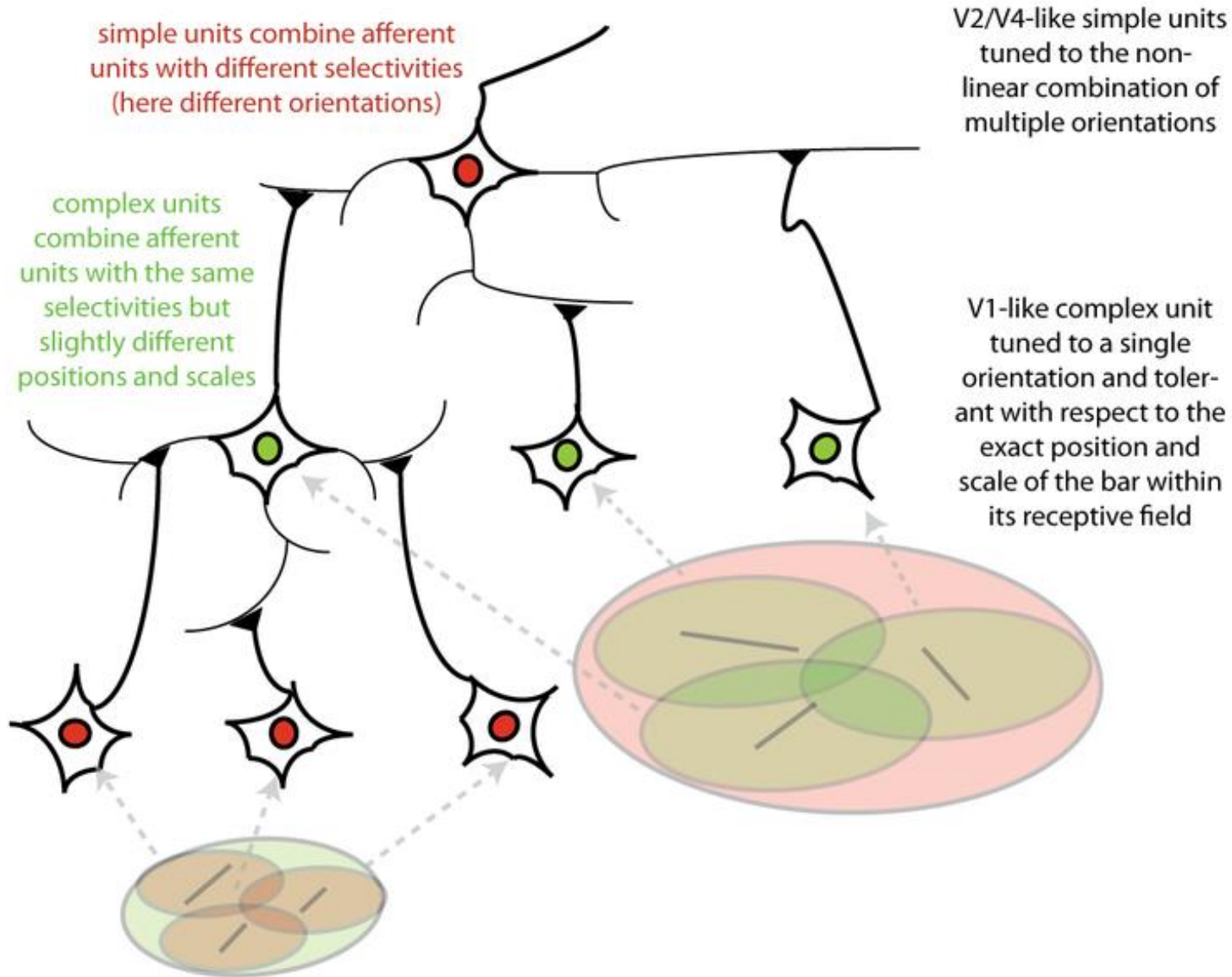
- Vision



dorsal stream: “where”

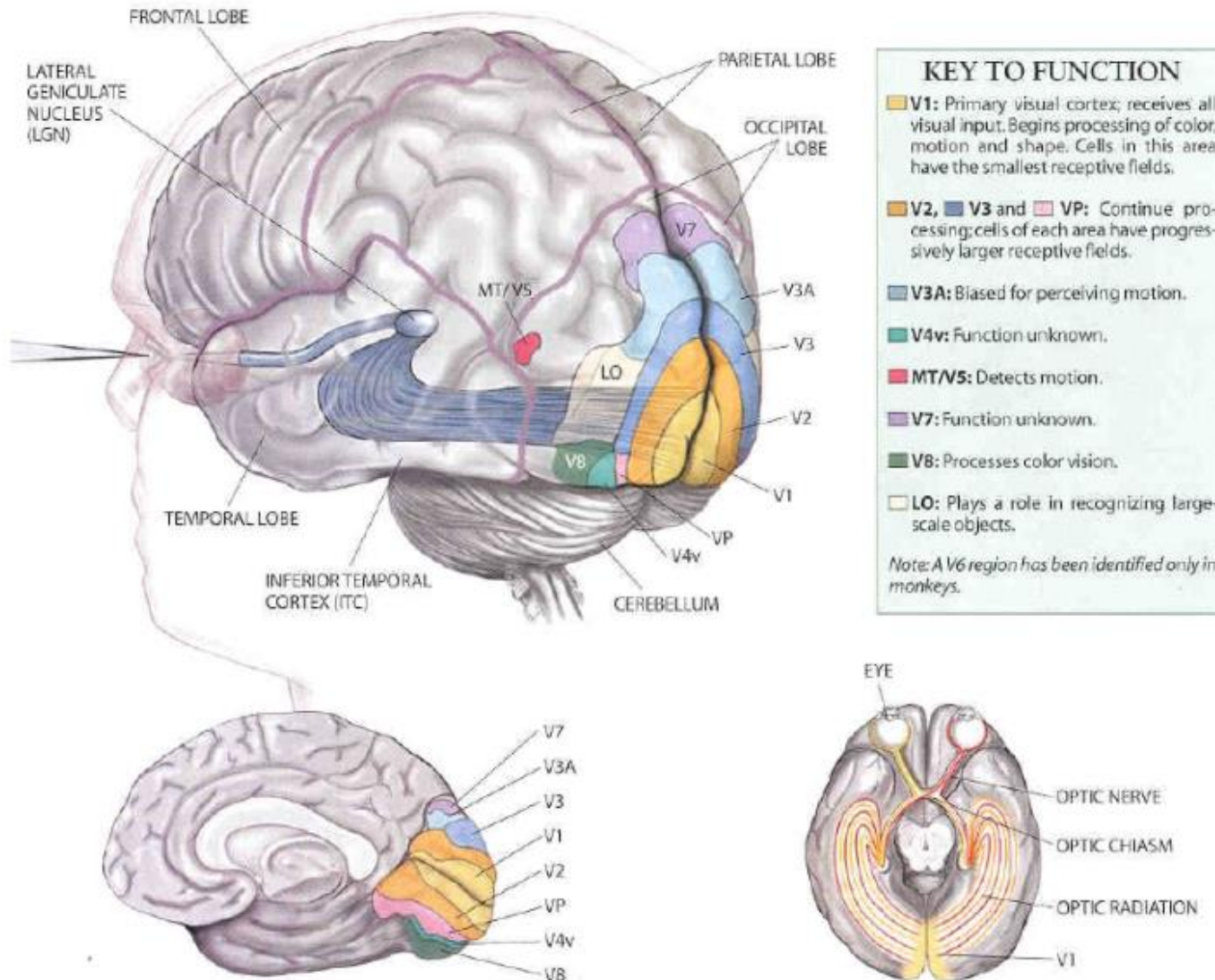
ventral stream: “what”

Visual representation

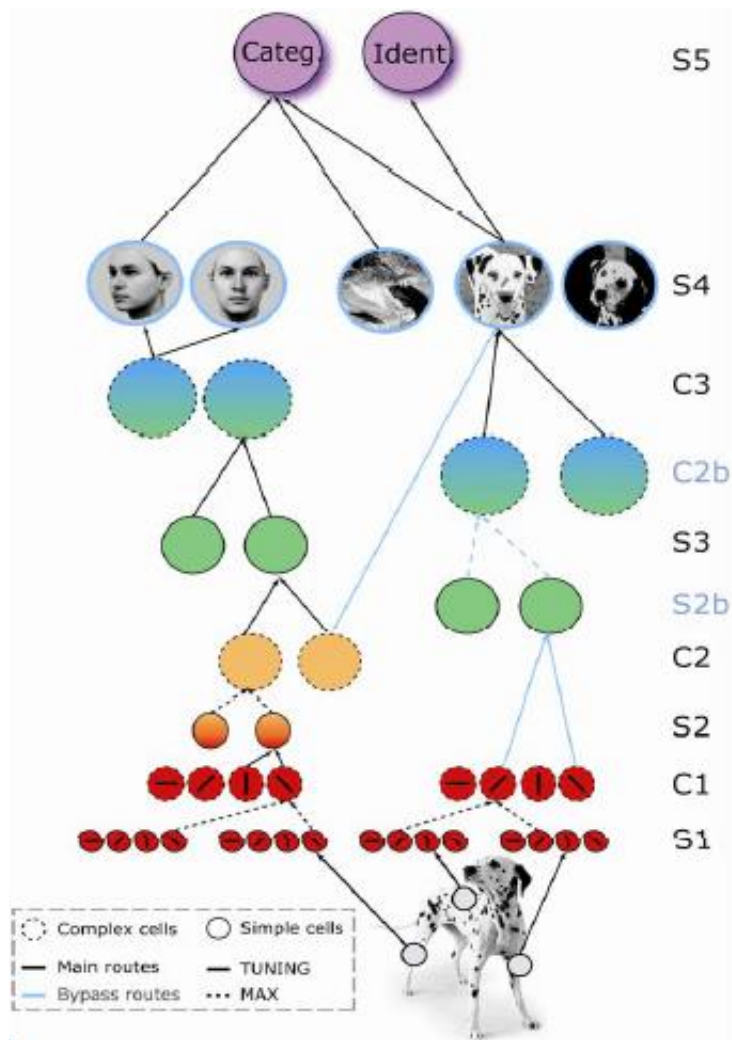


Visual cortical areas- human

































(N. K. Logothetis, "Vision: A window on consciousness", Scientific American, 1999)



HMax Framework (Serre et al.)



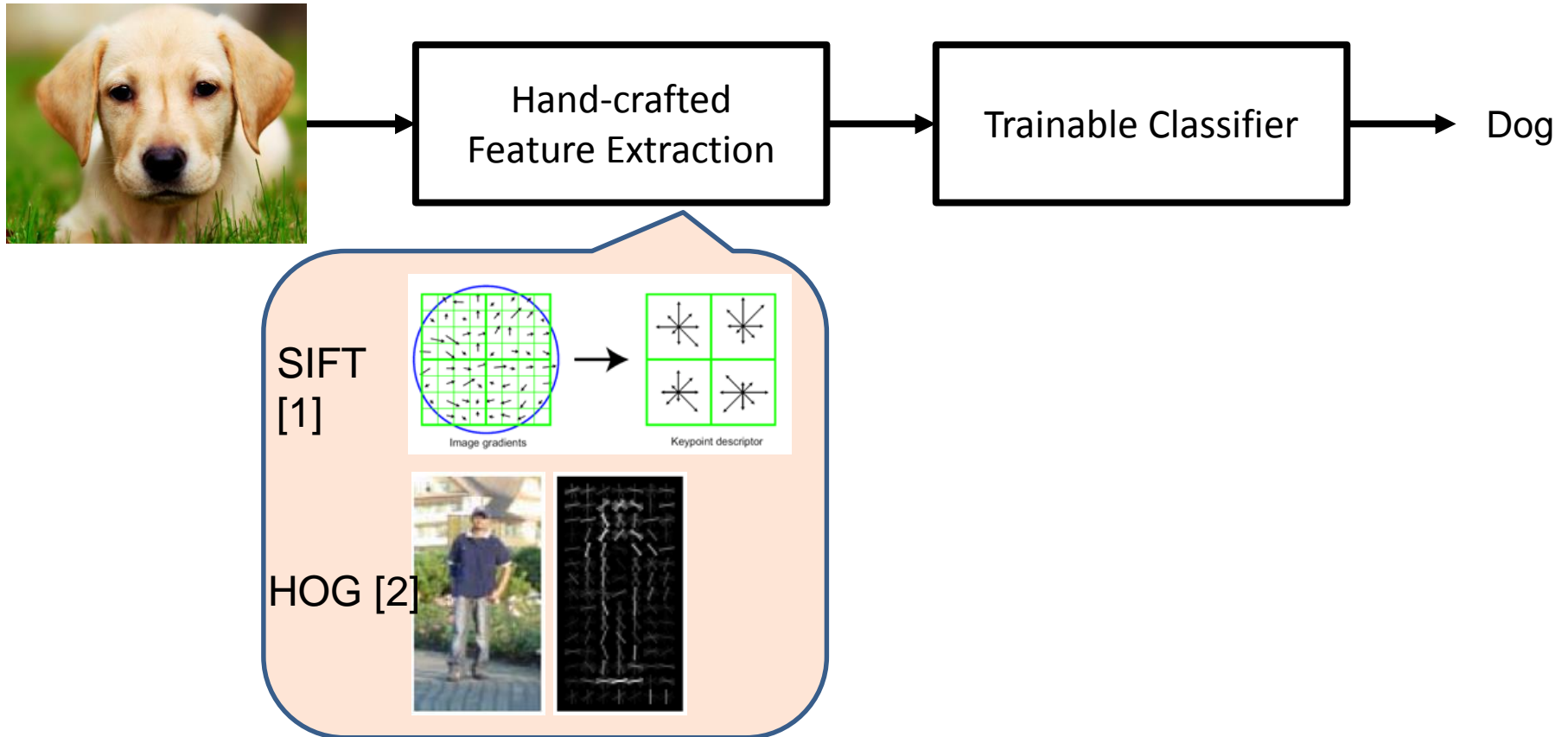
Serre, Oliva, and Poggio 2007

V2	V4	posterior IT	anterior IT
 	 	 	 
 	 	 	 
 	 	 	 
 	 	 	 

Kobatake and Tanaka, 1994

Motivation

- Make feature representation learnable instead of hand-crafting it.



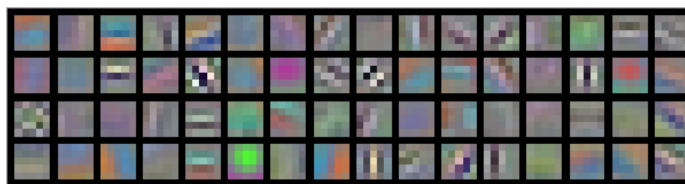
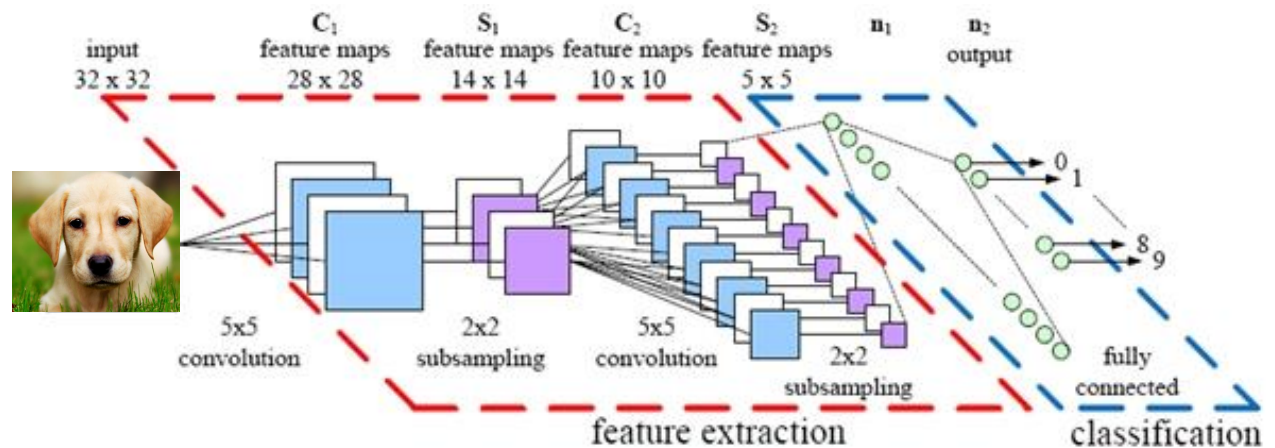
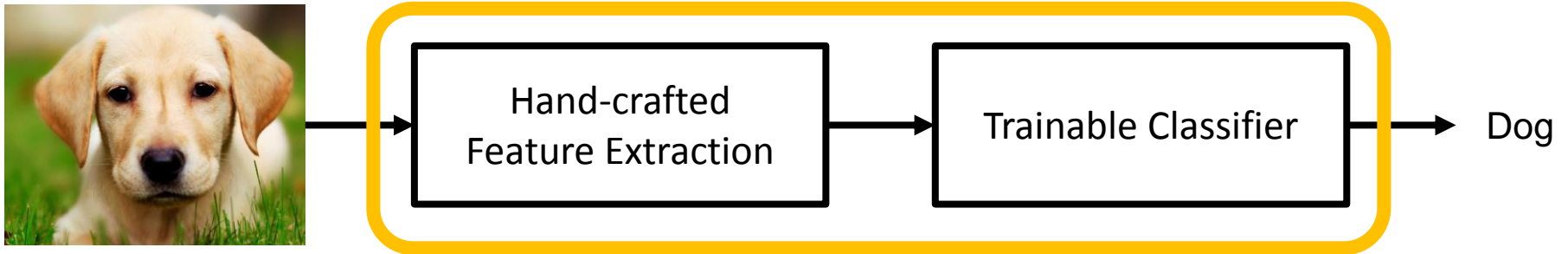
[1] Lowe, David G. "Object recognition from local scale-invariant features". ICCV 1999

[2] Dalal, N. and Triggs, B. "Histograms of oriented gradients for human detection". CVPR 2005

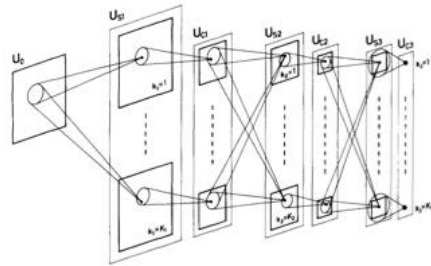
[3] <https://code.google.com/p/cuda-convnet/>

Motivation

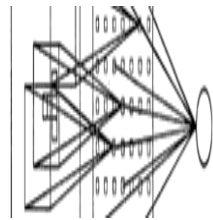
- Make feature representation learnable instead of hand-crafting it.



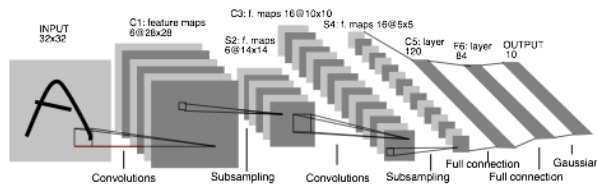
History of ConvNets



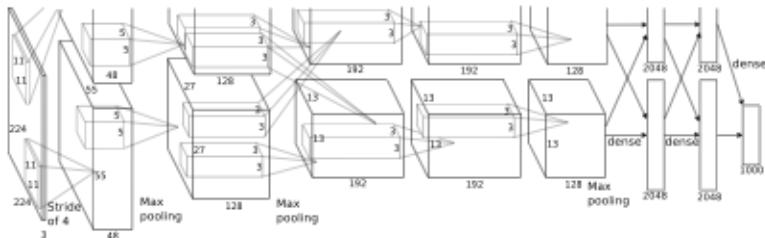
Fukushima 1980
Neocognitron



Rumelhart, Hinton, Williams 1986
“T” versus “C” problem



LeCun et al. 1989-1998
Hand-written digit reading



Krizhevsky, Sutskever, Hinton 2012
ImageNet classification breakthrough
“SuperVision” CNN

Problem of Current CNN

- Current CNN architecture is mostly based on the one developed in 1998.
- Hidden layers of CNN lack transparency during training.
- Exploding and vanishing gradients presence during back propagation training [1,2].

[1] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTAT, 2010.

[2] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In arXiv:1211.5063v2, 2014.

Deeply-Supervised Nets

To boost the classification performance by focusing three aspects:

- Robustness and discriminativeness of the learned features.
- Transparency of the hidden layers on the overall classification.
- Training difficulty due to the “exploding” and “vanishing” gradients.

Some Definitions

Input training set: $S = \{(X_i, y_i), i = 1..N\}$

$$X_i \in \mathbb{R}^n, y_i \in \{1..K\}$$

Recursive functions:

Features: $Z^{(m)} = f(Q^{(m)})$, and $Z^{(0)} \equiv X$

$$Q^{(m)} = W^{(m)} * Z^{(m-1)}$$

Summarize all the parameters as:

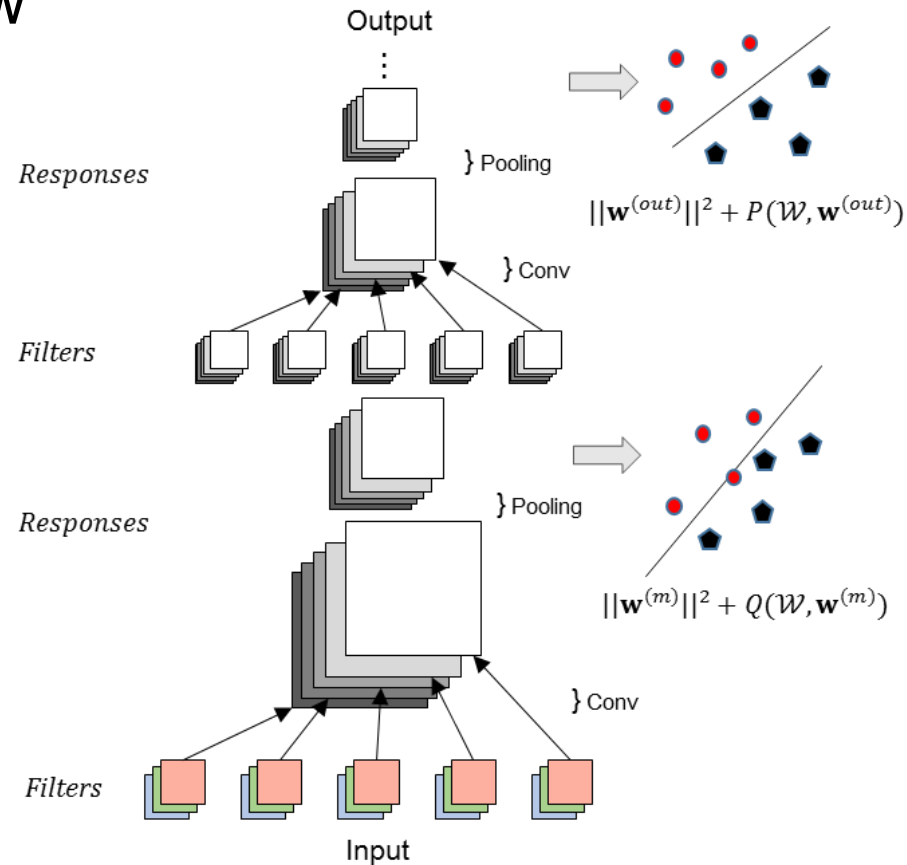
$$W = (W^{(1)}, \dots, W^{(out)})$$

In addition, we have SVM weights (to be discarded after training)

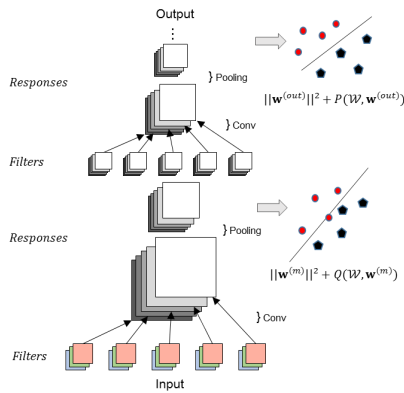
$$\mathbf{w} = (\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(M-1)})$$

Proposed Method

- Deeply-Supervised Nets (DSN)
- Direct supervision to intermediate layers to learn weights W, w



Formulations



standard objective function for SVM

Hidden layer supervision

$$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(W, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+$$

$$\mathcal{L}(W, \mathbf{w}^{(out)}) = \sum_{y_k \neq y} [1 - \langle \mathbf{w}^{(out)}, \phi(\mathbf{Z}^{(M)}, y) - \phi(\mathbf{Z}^{(M)}, y_k) \rangle]_+^2$$

$$\ell(W, \mathbf{w}^{(m)}) = \sum_{y_k \neq y} [1 - \langle \mathbf{w}^{(m)}, \phi(\mathbf{Z}^{(m)}, y) - \phi(\mathbf{Z}^{(m)}, y_k) \rangle]_+^2$$

Multi-class hinge loss between responses Z and true label y

Formulations

- The gradient of the objective function w.r.t the weights:

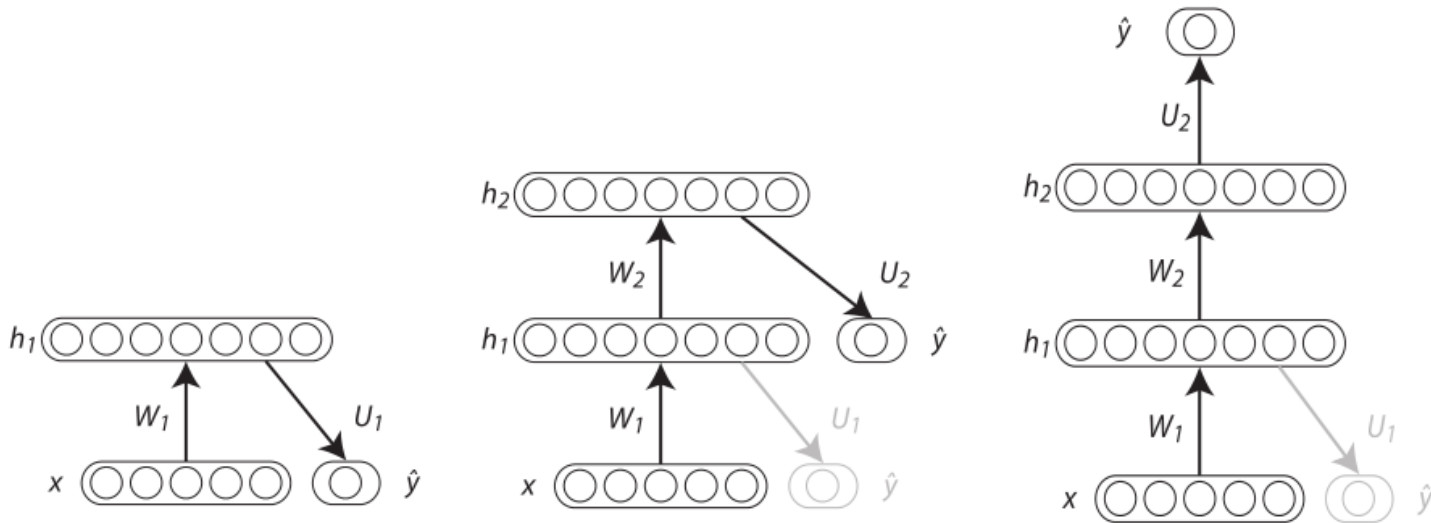
$$\frac{\partial F}{\partial \mathbf{w}^{(out)}} = 2\mathbf{w}^{(out)} - 2 \sum_{y_k \neq y} [\phi(\mathbf{Z}^{(M)}, y) - \phi(\mathbf{Z}^{(M)}, y_k)] [1 - \langle \mathbf{w}^{(out)}, \phi(\mathbf{Z}^{(M)}, y) - \phi(\mathbf{Z}^{(M)}, y_k) \rangle]_+$$

$$\frac{\partial F}{\partial \mathbf{w}^{(m)}} = \begin{cases} 0, & \text{when } \|\mathbf{w}^{(m)}\|^2 + \ell(\mathbf{W}, \mathbf{w}^{(m)}) \leq \gamma \\ \alpha_m \left\{ 2\mathbf{w}^{(m)} - 2 \sum_{y_k \neq y} [\phi(\mathbf{Z}^{(m)}, y) - \phi(\mathbf{Z}^{(m)}, y_k)] [1 - \langle \mathbf{w}^{(m)}, \phi(\mathbf{Z}^{(m)}, y) - \phi(\mathbf{Z}^{(m)}, y_k) \rangle]_+ \right\}, & \text{otherwise.} \end{cases}$$

- Apply the computed gradients to perform stochastic gradient descend and then iteratively train our DSN model.

Greedy layer-wise supervised pretraining

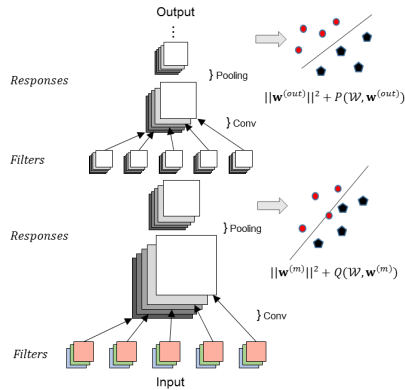
(Bengio et al. 2007)



Essentially shown to be ineffective (worse than unsupervised pre-training).

	Experiment 2			Experiment 3		
	train.	valid.	test	train.	valid.	test
DBN, unsupervised pre-training	0%	1.2%	1.2%	0%	1.5%	1.5%
Deep net, auto-associator pre-training	0%	1.4%	1.4%	0%	1.4%	1.6%
Deep net, supervised pre-training	0%	1.7%	2.0%	0%	1.8%	1.9%
Deep net, no pre-training	.004%	2.1%	2.4%	.59%	2.1%	2.2%
Shallow net, no pre-training	.004%	1.8%	1.9%	3.6%	4.7%	5.0%

Deeply-supervised nets

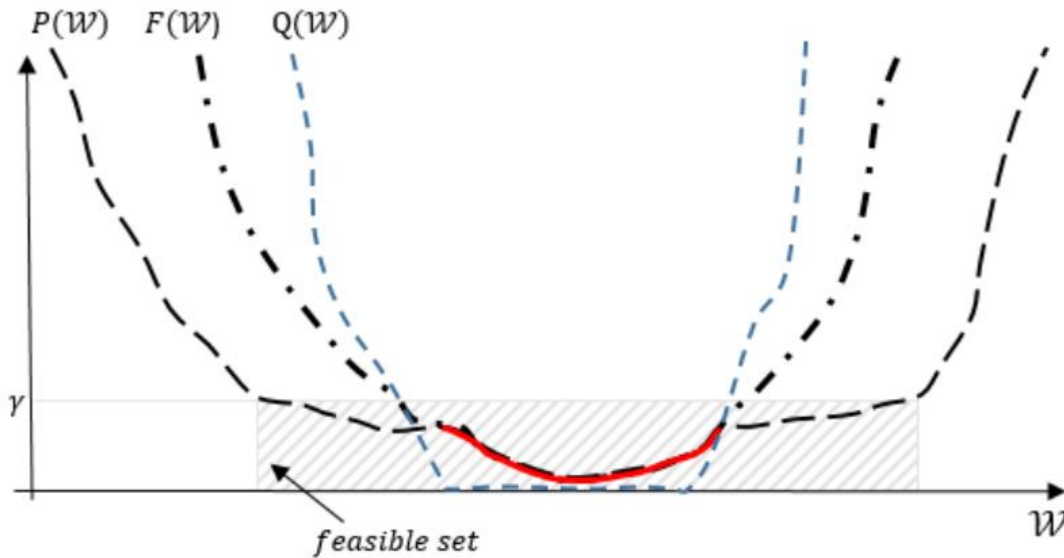


$$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(W, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+$$

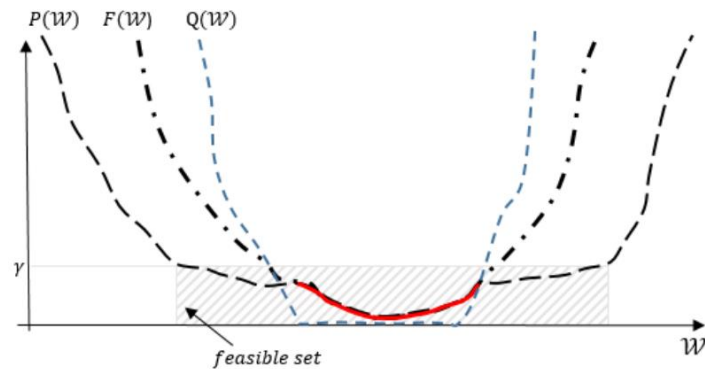
$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

$$\mathcal{P}(W) \equiv \|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(W, \mathbf{w}^{(out)})$$

$$\mathcal{Q}(W) \equiv \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+$$



With a loose assumption



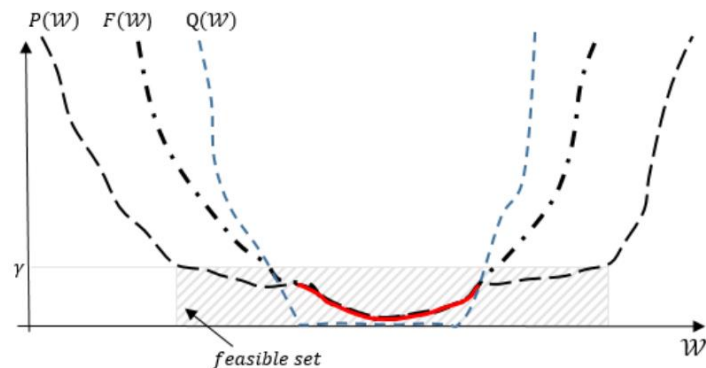
$$F(W') \geq F(W) + \langle \mathbf{g}, W' - W \rangle + \frac{\lambda}{2} \|W' - W\|^2$$

$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

Definition We denote by $\mathcal{S}_\gamma(F) = \{W \mid F(W) \leq \gamma\}$ the γ -sublevel set, stated here for the function $F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$.

Lemma 1 $\forall m, m' = 1 \dots M - 1$, and $m' > m$ if $\|\mathbf{w}^{(m)}\|^2 + \ell((\hat{W}^{(1)}, \dots, \hat{W}^{(m)}), \mathbf{w}^{(m)}) \leq \gamma$ then there exists $(\hat{W}^{(1)}, \dots, \hat{W}^{(m)}, \dots, \hat{W}^{(m')})$ such that $\|\mathbf{w}^{(m')}\|^2 + \ell((\hat{W}^{(1)}, \dots, \hat{W}^{(m)}, \dots, \hat{W}^{(m')}), \mathbf{w}^{(m')}) \leq \gamma$. \square

With a loose assumption



$$F(W') \geq F(W) + \langle \mathbf{g}, W' - W \rangle + \frac{\lambda}{2} \|W' - W\|^2$$

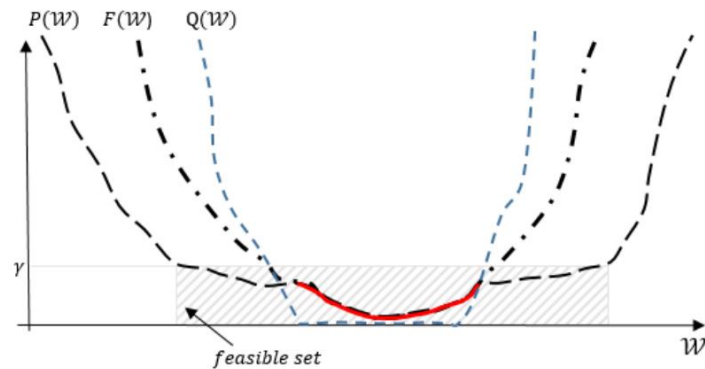
$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

Lemma 2 Suppose $\mathbb{E}[\|\hat{\mathbf{g}}_{\mathbf{p}_t}\|^2] \leq G^2$ and $\mathbb{E}[\|\hat{\mathbf{g}}_{\mathbf{q}_t}\|^2] \leq G^2$, and we use the update rule of $W_{t+1} = \Pi_{\mathcal{W}}(W_t - \eta_t(\hat{\mathbf{g}}_{\mathbf{p}_t} + \hat{\mathbf{g}}_{\mathbf{q}_t}))$ where $\mathbb{E}[\hat{\mathbf{g}}_{\mathbf{p}_t}] = \mathbf{g}_{\mathbf{p}_t}$ and $\mathbb{E}[\hat{\mathbf{g}}_{\mathbf{q}_t}] = \mathbf{g}_{\mathbf{q}_t}$. If we use $\eta_t = 1/(\lambda_1 + \lambda_2)t$, then at iteration T

$$\mathbb{E}[\|W_T - W^*\|^2] \leq \frac{4G^2}{(\lambda_1 + \lambda_2)^2 T}$$

Based on Lemma 1 in: A. Rakhlin, O. Shamir, and K. Sridharan. “Making gradient descent optimal for strongly convex stochastic optimization”. ICML, 2012.

A loose assumption



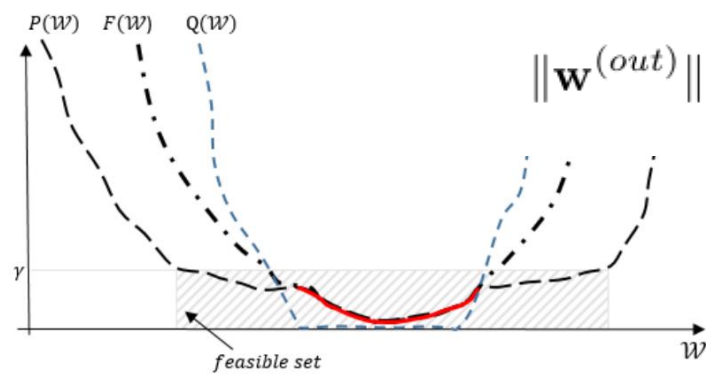
$$F(W') \geq F(W) + \langle \mathbf{g}, W' - W \rangle + \frac{\lambda}{2} \|W' - W\|^2$$

$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

Lemma 3 We follow the assumptions in lemma [2](#), with the exception that we assume $\eta_t = 1/t$ since λ_1 and λ_2 are not always readily available; as we discuss in the appendix, we also expect the combined λ to be small. When we begin in the region $\|W_1 - W^*\|^2 \leq D$, the convergence rate is bounded by

$$\mathbb{E}[\|W_T - W^*\|^2] \leq e^{-2\lambda(\ln(T-1) + 0.578)} D + 4G^2 \sum_{t=1}^{T-1} \frac{1}{t^2} \left(\frac{t}{T-1}\right)^{2\lambda}$$

A loose assumption

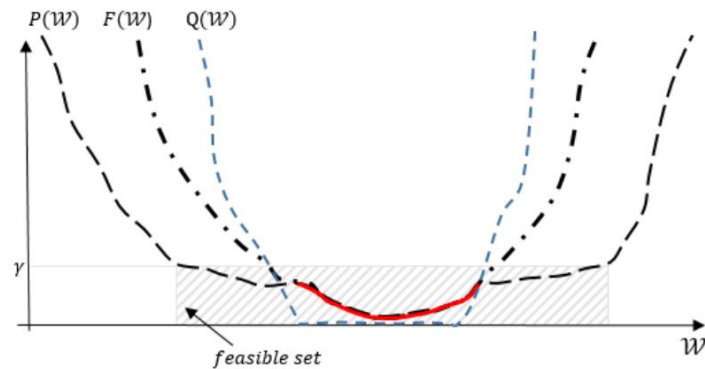


$$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(W, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+$$

$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

Theorem 1 Let $\mathcal{P}(W)$ be λ_1 -strongly convex and $\mathcal{Q}(W)$ be λ_2 -strongly convex near optimal W^* and denote by $W_T^{(F)}$ and $W_T^{(P)}$ the solution after T iterations when following SGD on $F(W)$ and $\mathcal{P}(W)$, respectively. Then DSN framework improves the relative convergence speed $\frac{\mathbb{E}[\|W_T^{(P)} - W^*\|^2]}{\mathbb{E}[\|W_T^{(F)} - W^*\|^2]}$, viewed from the ratio of their upper bounds as $\Theta\left(\frac{(\lambda_1 + \lambda_2)^2}{\lambda_1^2}\right)$, when $\eta_t = 1/\lambda t$.

A loose assumption



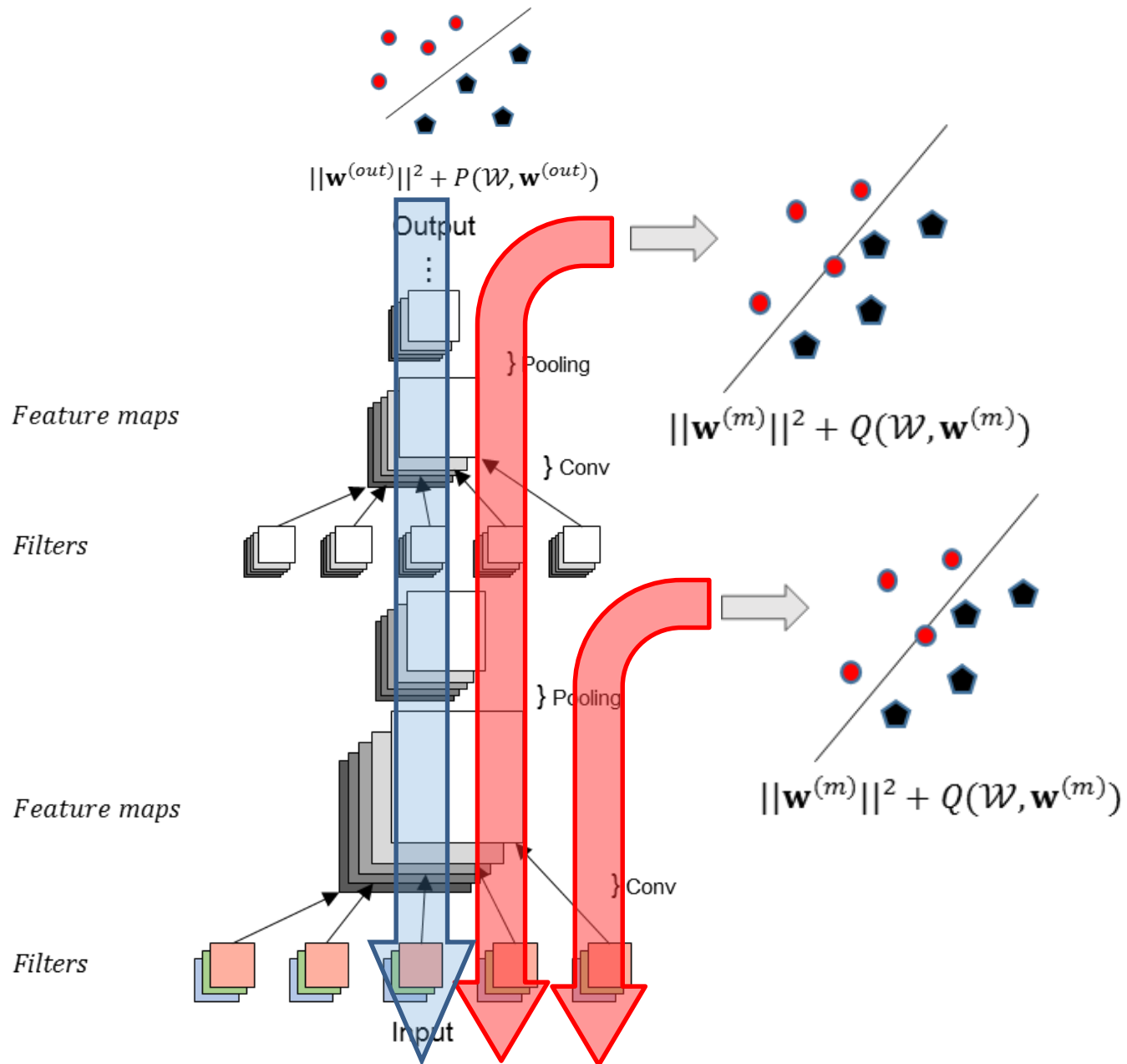
$$F(W') \geq F(W) + \langle \mathbf{g}, W' - W \rangle + \frac{\lambda}{2} \|W' - W\|^2$$

$$F(W) \equiv \mathcal{P}(W) + \mathcal{Q}(W)$$

$$\mathbb{E}[\|W_T - W^*\|^2] \leq e^{-2\lambda(\ln(T-1) + 0.578)} D + 4G^2 \sum_{t=1}^{T-1} \frac{1}{t^2} \left(\frac{t}{T-1}\right)^{2\lambda}$$

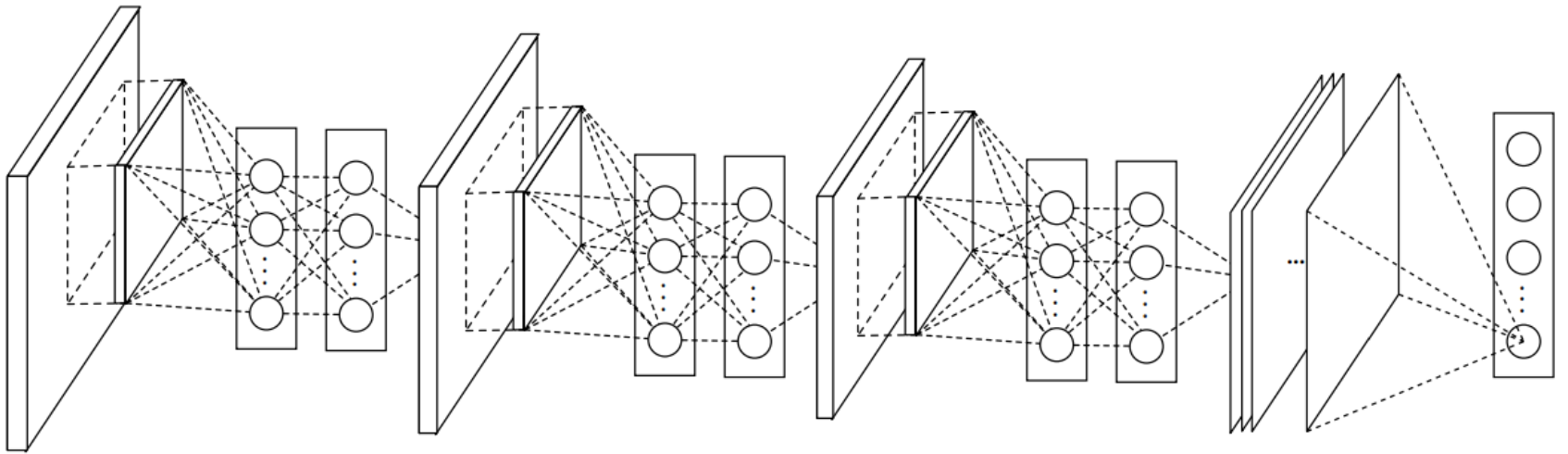
Remark When $\eta_t = 1/t$, T is large, and the first term dominates, the upper bound ratio for $\frac{\mathbb{E}[\|W_T^{(\mathcal{P})} - W^*\|^2]}{\mathbb{E}[\|W_T^{(F)} - W^*\|^2]}$ is roughly at the order of $\Theta(e^{2\ln(T)\lambda_2})$. If the second term dominates, the convergence of $F(W)$ over $\mathcal{P}(W)$ is also advantageous with the ratio at an order of $\Theta(e^{2\ln((T-1)/(T-2))\lambda_2})$ loosely.

Illustration



Network-in-Network

(M. Lin, Q. Chen, and S. Yan, ICLR 2014)



Some alternative formulations

1. Constrained optimization:

$$\begin{aligned} & \text{minimize } \|w^{(out)}\|^2 + \mathcal{L}(W, w^{(out)}) \\ & \text{subject to } \|w^{(m)}\|^2 + \ell(W, w^{(m)}) \leq \gamma, m = 1..M - 1 \end{aligned}$$

2. Fixed $\alpha(m)$:

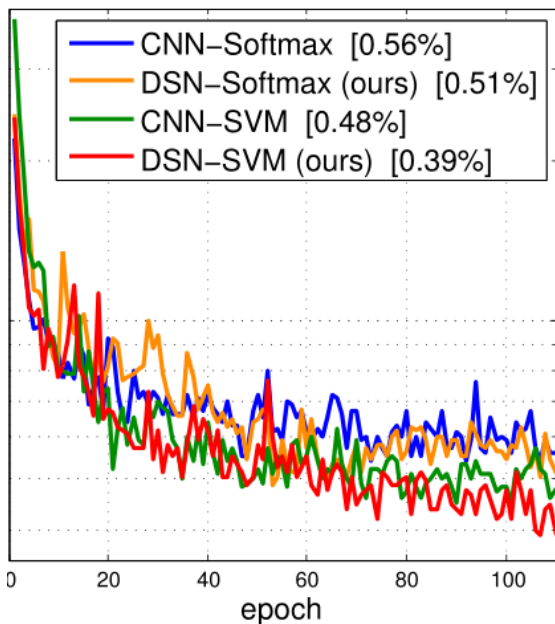
$$\text{minimize } \|w^{(out)}\|^2 + \mathcal{L}(W, w^{(out)}) + \sum_{m=1}^{M-1} \alpha_m \left| \|w^{(m)}\|^2 + \ell(W, w^{(m)}) - \gamma \right|_+$$

3. Decay function for $\alpha(m)$:

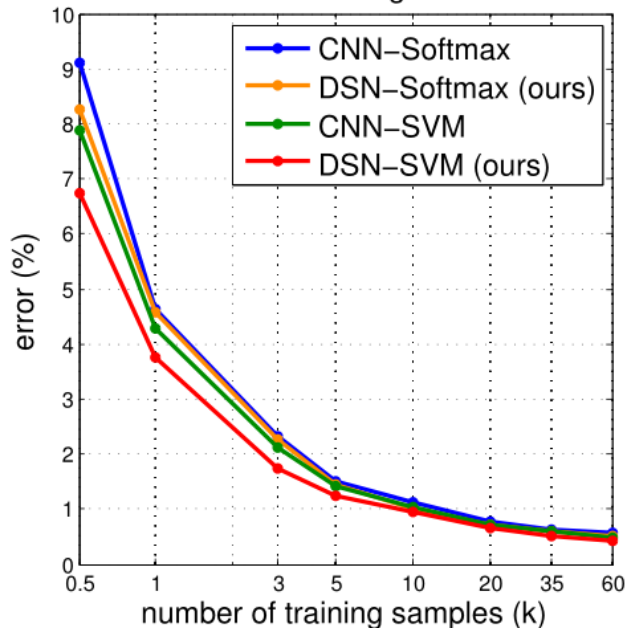
$$\begin{aligned} & \text{minimize } \|w^{(out)}\|^2 + \mathcal{L}(W, w^{(out)}) + \sum_{m=1}^{M-1} \alpha_m \left| \|w^{(m)}\|^2 + \ell(W, w^{(m)}) - \gamma \right|_+ \\ & \alpha(m) \equiv c(m) \left(1 - \frac{t}{N} \right) \end{aligned}$$

Experiment on the MNIST dataset

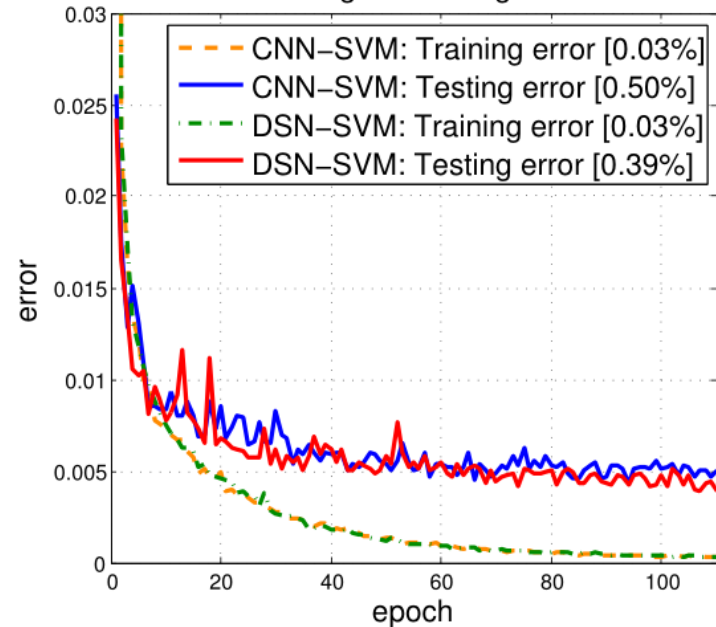
MNIST testing error



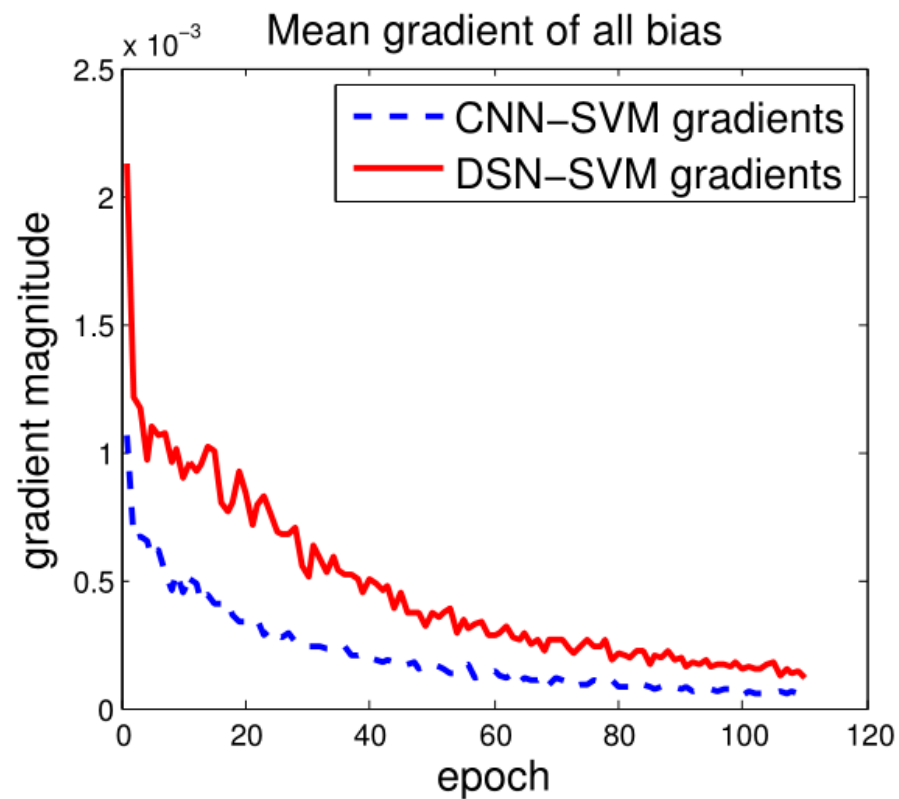
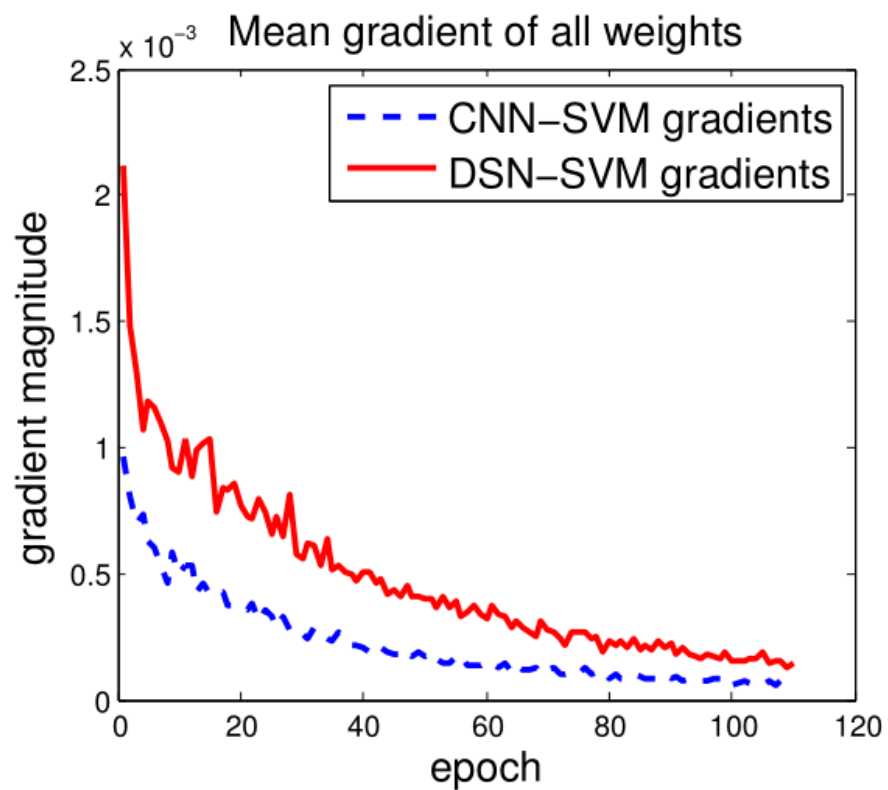
MNIST testing error



Training and testing error



Some empirical results



Experiment on the MNIST dataset

Method	Error Rate (%)
CNN	0.53
Stochastic Pooling	0.47
Network in Network	0.47
Maxout Network	0.45
CNN (layer-wise pre-training)	0.43
DSN (ours)	0.39

- **CNN:** Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. “Backpropagation applied to handwritten zip code recognition”. Neural Computation, 1989.
- **Stochastic Pooling:** M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. ICLR, 2013.
- **Network in Network:** M. Lin, Q. Chen, and S. Yan. Network in network. ICLR, 2014.
- **Maxout Network:** I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. ICML, 2013.

MNIST training details

layer	details
conv1	stride 2, kernel 5x5, relu, channel_output 32
+ L2SVM	input conv1 (after max pooling), squared hinge loss
conv2	stride 2, kernel 5x5, relu, channel_output 64
+ L2SVM	input conv2 (after max pooling), squared hinge loss
fc3	relu, channel_output 500, dropout rate 0.5
fc4	channel_output 10
Output layer: L2SVM	squared hinge loss

110 epochs

$$Q(W) \equiv \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+$$

- Base learning rate = 0.4.
- $\alpha_m = 0.1 \times \left(1 - \frac{t}{N}\right)$

CIFAR results

CIFAR-10 classification error.

Method	Error(%)
No Data Augmentation	
Stochastic Pooling	15.13
Maxout Networks	11.68
Network in Network (NIN)	10.41
NIN (layer-wise pre-training)	9.92
DSN (ours)	9.69
With Data Augmentation	
Maxout Networks	9.38
Dropconnect	9.32
Network in Network	8.81
DSN (ours)	7.97

CIFAR-100 classification error.

Method	Error(%)
Stochastic Pooling	42.51
Maxout Networks	38.57
Tree based Priors	36.85
Network in Network	35.68
DSN (ours)	34.57

- **Stochastic Pooling:** M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. ICLR, 2013.
- **Maxout Network:** I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. ICML, 2013.
- **Network in Network:** M. Lin, Q. Chen, and S. Yan. Network in network. ICLR, 2014.
- **Dropconnect:** W. Li, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. ICML, 2013.
- **Tree based Priors:** N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. NIPS, 2013.

DSN on CIFAR-10 training details

layer	details
conv1	stride 2, kernel 5x5, channel_output 192
+ L2SVM	input conv1 (before relu), squared hinge loss
2 NIN layers	1x1 conv, channel_output 160, 96, dropout 0.5
conv2	stride 2, kernel 5x5, channel_output 192
+ L2SVM	input conv2 (before relu), squared hinge loss
2 NIN layers	1x1 conv, channel_output 192, 192, dropout rate 0.5
conv3	stride 1, kernel 3x3, relu, channel_output 192
+ L2SVM	input conv3 (before relu), squared hinge loss
2 NIN layers	1x1 conv, channel_output 192, 10, dropout rate 0.5 global average pooling
Output layer: L2SVM	input global average pooling, squared hinge loss

400 epochs

$$Q(W) \equiv \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+.$$

- Base learning rate = 0.025, reduce learning rate twice by a factor of 20.
- $\alpha_m = 0.001$ fixed for all companion objectives.
- The companion objectives vanish after 100 epochs $\equiv \gamma(0.8, 0.8, 1.4)$ for each layer,

DSN on CIFAR-100 training details

layer	details
conv1	stride 2, kernel 5x5, channel_output 192
+ SOFTMAX	input conv1 (before relu), softmax loss
2 NIN layers	1x1 conv, channel_output 160, 96, dropout 0.5
conv2	stride 2, kernel 5x5, channel_output 192
+ SOFTMAX	input conv2 (before relu), softmax loss
2 NIN layers	1x1 conv, channel_output 192, 192, dropout rate 0.5
conv3	stride 1, kernel 3x3, relu, channel_output 192
+ SOFTMAX	input conv3 (before relu), softmax loss
2 NIN layers	1x1 conv, channel_output 192, 10, dropout rate 0.5 global average pooling
Output layer: SOFTMAX	input global average pooling, softmax loss

400 epochs

$$Q(W) \equiv \sum_{m=1}^{M-1} \alpha_m [\|w^{(m)}\|^2 + \ell(W, w^{(m)}) - \gamma]_+.$$

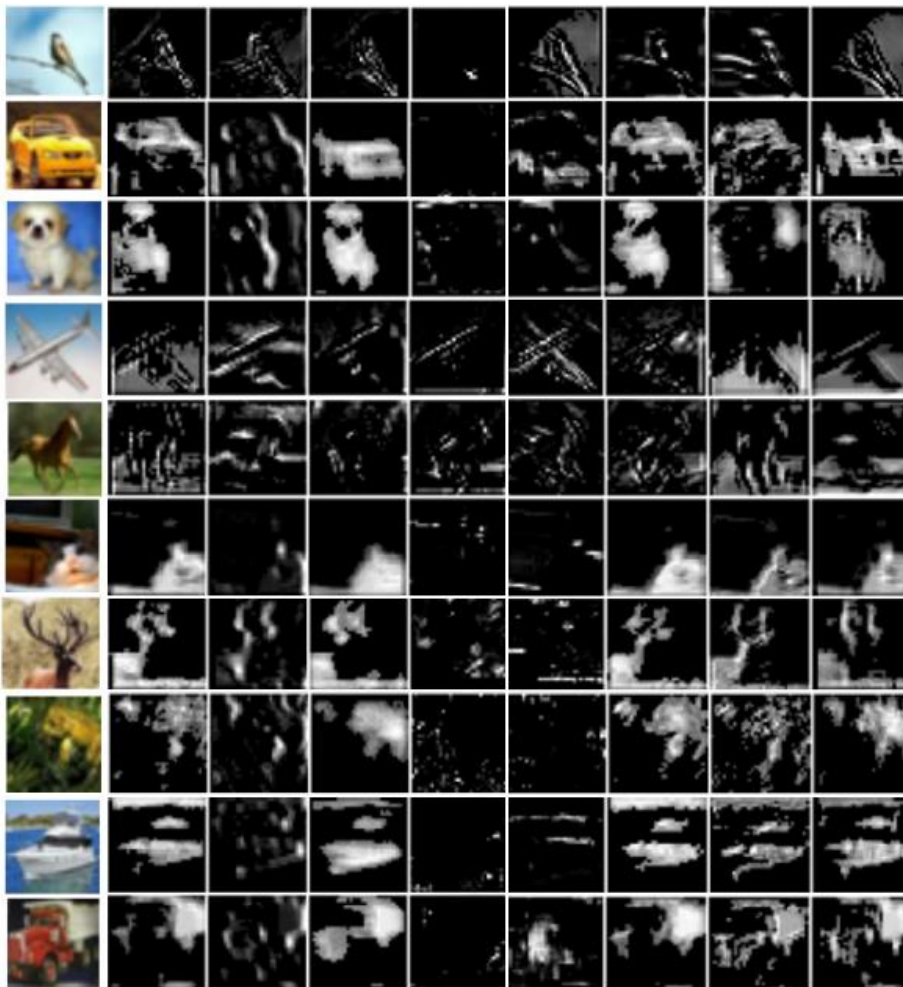
- Hyper-parameters and epoch schedules are identical to those in CIFAR-10
- The only difference is using Softmax classifiers instead of L2SVM classifiers

Result on the SVHN dataset

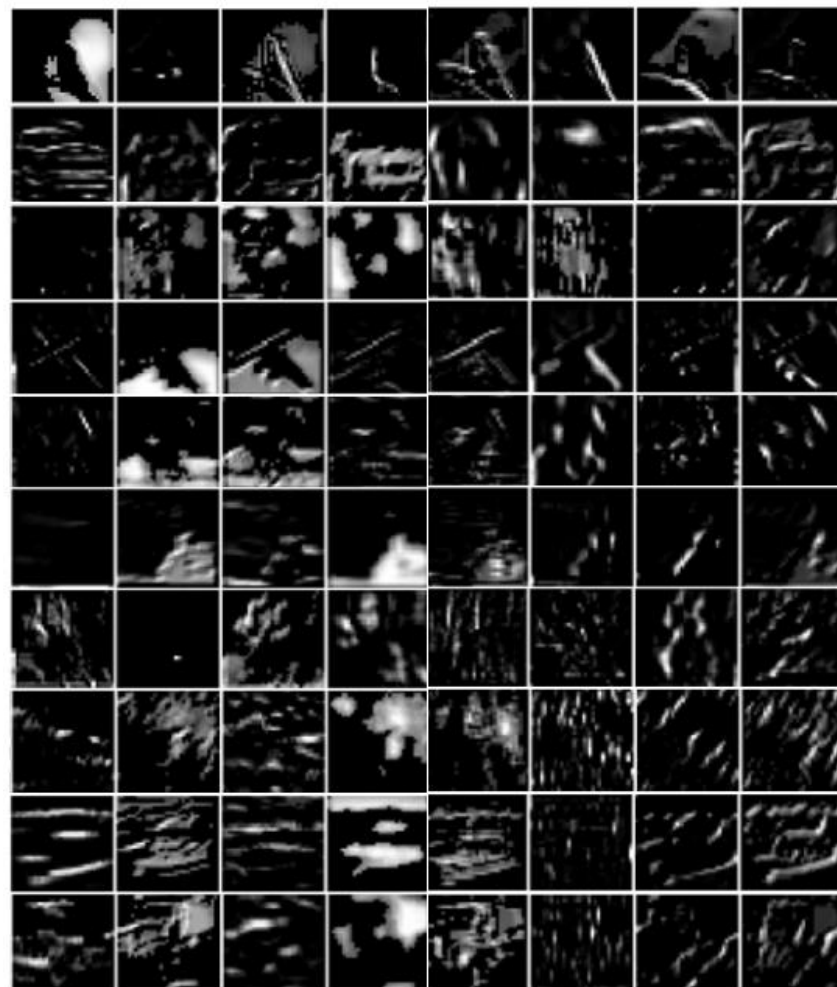
Method	Error(%)
Stochastic Pooling	2.80
Maxout Networks	2.47
Network in Network	2.35
Dropconnect	1.94
DSN (ours)	1.92

- **Stochastic Pooling:** M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. ICLR, 2013.
- **Maxout: Network:** I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. ICML, 2013.
- **Network in Network:** M. Lin, Q. Chen, and S. Yan. Network in network. ICLR, 2014.
- **Dropconnect:** W. Li, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. ICML, 2013.

Visualization of learned features



DSN



CNN

ImageNet



GoogLeNet

GoogLeNet: C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. “Going deeper with convolutions”. arXiv:1409.4842, 2014.

Results on ImageNet

Method	top-1 val. error(%)	top-5 val. error(%)
CNN 8-layer	40.7	18.2
DSN 8-layer (ours)	39.6	17.8
CNN 11-layer	34.5	13.9
DSN 11-layer (ours)	33.6	13.1

DSN on ImageNet 2012 training details

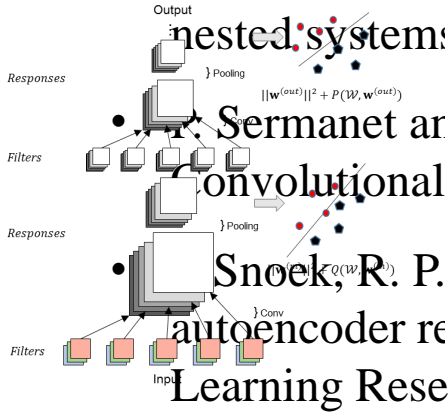
layer	details
conv1	stride 4, kernel 11x11, relu, channel_output 64
conv2	stride 1, kernel 5x5, relu, channel_output 192
conv3	stride 1, kernel 3x3, relu, channel_output 384
conv4	stride 1, kernel 3x3, relu, channel_output 256
+ SOFTMAX	softmax loss
conv5	stride 1, kernel 3x3, relu, channel_output 256
fc6	channel_output 4096, dropout rate 0.5
fc7	channel_output 4096, dropout rate 0.5
fc8	channel_output 1000
Output layer	softmax loss

N=90 epochs

- Base learning rate = 0.01 with decay factor 0.1. Learning rate is decayed whenever validation error stop decreasing until it reaches 10^{-5}
- The companion objectives are weighted by $\alpha = 0.4$ with decay factor = $\left(1 - \frac{t}{N}\right)$ where t is current epoch index and N is the number of total epoch.

Relation to prior work

- M. A. Carreira-Pernin and W. Wang. "Distributed optimization of deeply nested systems: Sermanet and LeCun 2011"



$$\| \mathbf{w}^{(out)} \|^2 + \mathcal{L}(W, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\| \mathbf{w}^{(m)} \|^2 + \ell(W, \mathbf{w}^{(m)}) - \gamma]_+,$$

Sermanet and LeCun 2011

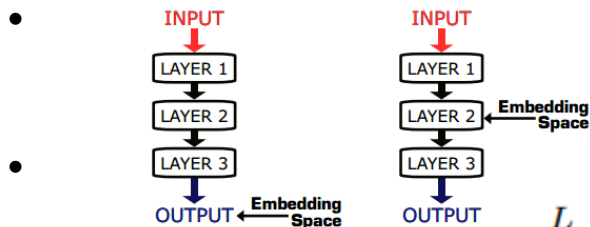
$$\mathcal{L}(W, \mathbf{w}^{(out)}) = \sum_{y_k \neq y} [1 - \langle \mathbf{w}^{(out)}, \phi(\mathbf{Z}^{(M)}, y) - \phi(\mathbf{Z}^{(M)}, y_k) \rangle]_+^2$$

Snoek, R. P. and Adams, 2012

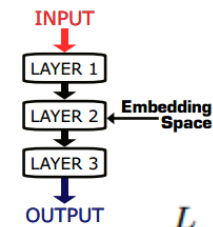
$$\ell(W, \mathbf{w}^{(m)}) = \sum_{y_k \neq y} [1 - \langle \mathbf{w}^{(m)}, \phi(\mathbf{Z}^{(m)}, y) - \phi(\mathbf{Z}^{(m)}, y_k) \rangle]_+^2$$

Learning Research, 2012.

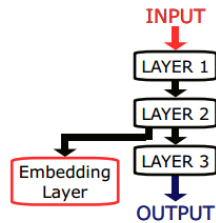
DSN



(a) Output



(b) Internal



(c) Auxiliary

learning via semi-supervised embedding".

$$\sum_{i=1}^L \ell(f(x_i), y_i) + \lambda \sum_{i,j=1}^{L+U} L(f^k(x_i), f^k(x_j), W_{ij})_{K+1}$$

$$\alpha) L_{\text{auto}}(\phi, \psi) + \alpha L_{\text{GP}}(\phi, \psi) \quad \left. \begin{matrix} \text{Weston et al, 2008.} \\ \text{Snoek and Adams, 2012} \end{matrix} \right\} n = 1, \dots, N.$$

architecture. The input is processed in a feed-forward of convolutions and subsampling, and classifier. The output of the 1st stage is also higher-resolution features.

Figure 1. Three modes of embedding in deep architectures.

Sermanet and LeCun 2011

Conclusions of DSN

- For relatively shallow networks, DSN provides a strong regularization to reduce the test error.
- For very deep networks, DSN greatly relieves the vanishing gradient problem that makes the learning process otherwise very hard to train.
- We provide a new DL formulation and analysis, but many problems remain open.

Thank you!

Questions?