Outline for today

- Information
  - Inference variational autoencoders
  - mutual information and the information bottleneck
  - generalization vs. representation bounds
  - information plane illustration of: training, data size, weights, depth
- Dropout: regularization, test error, and sparsification

Mutual Information (MI) and entropy pt. 1

Recall the the Kullback-Leibler (KL) Divergence between distributions $p(x)$ and $q(x)$:

$$\begin{aligned} D_{KL}(p(x)||q(x)) &= \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \\ &= -\sum_x p(x)\log(q(x)) + \sum_x p(x)\log(p(x)) \\ &=: H(P,Q) - H(P) \end{aligned}$$

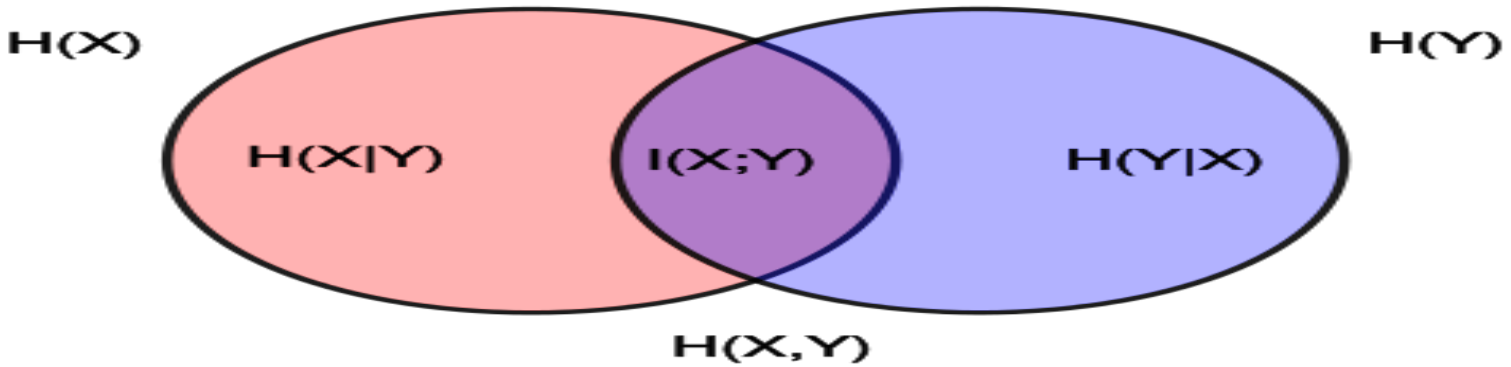
which is also in terms of the entropy $H(\cdot)$ of the distributions. Let $p(x,y)$ be the joint distribution and $p(x)$ and $p(y)$ the marginal probabilities; for instance $p(x) = \int_y p(x,y)dy$, then:

$$\begin{aligned} I(X;Y) &= D_{KL}(p(x,y)||p(x)p(y)) = \sum_{x,y} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\ &= \sum_{x,y} p(x,y)\log\left(\frac{p(x|y)}{p(x)}\right) = H(x) - H(X|Y) \end{aligned}$$

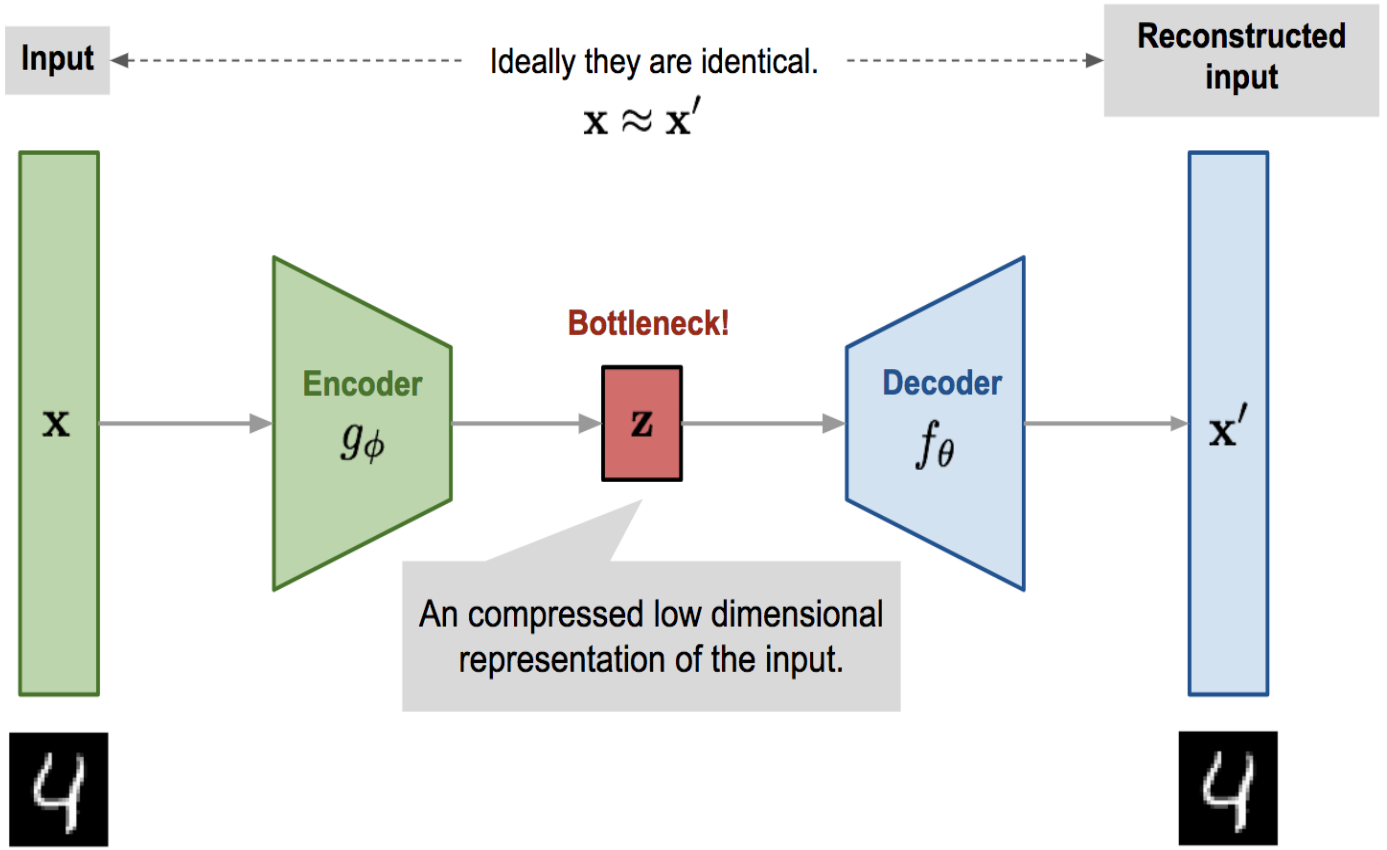
Mutual Information (MI) and entropy pt. 1

A few properties:

- $I(X;Y) \geq 0$, and equals 0 if and only if X and Y are independent, that is $p(x,y) = p(x)p(y)$.
- As $I(X;Y) \geq 0$, it follows from $I(X;Y) := H(X) - H(X|Y)$ that $H(X) \geq H(X|Y)$.
- The "data processing inequality" states that if $X \to Y \to Z$ is a Markov chain, then $I(X;Y) \geq I(X;Z)$ with equality if and only if $I(X;Y) = I(X;Z)$.



Autoencoder Illustration



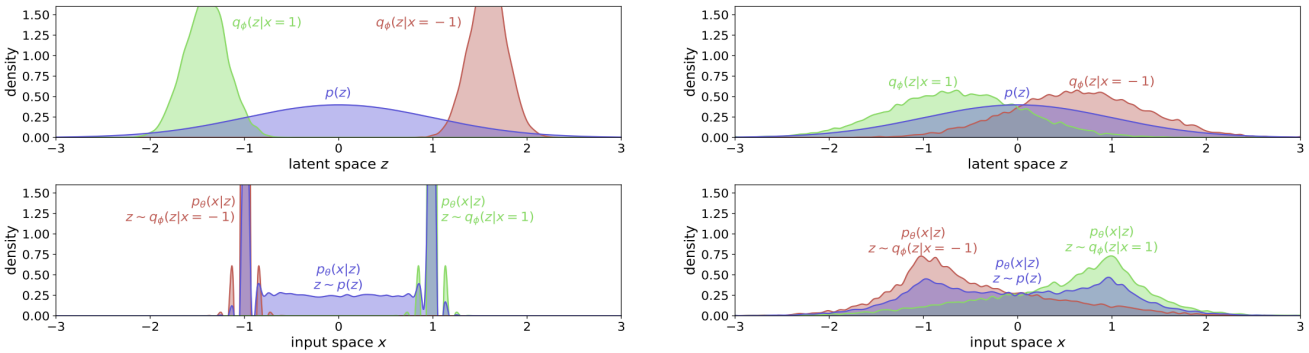
[1]

The parameters, (θ, ϕ), of the autoencoder are then learned:

$$\mathcal{L}(\theta, \phi) = n^{-1} \sum_{\mu=1}^{n} l(x_\mu, f_\theta(g_\phi(x_\mu)))$$

[1] https://lilianweng.github.io/lil-log/2018/08/12/

Inference Variational Autoencoders (Zhao et al. 17'[2])



ELBO InfoVAE ($\lambda = 500$)

Figure 1: Verification of Proposition 1 where the dataset only contains two examples $\{-1, 1\}$. **Top:** density of the distributions $q_\phi(z|x)$ when $x = 1$ (red) and $x = -1$ (green) compared with the true prior $p(z)$ (purple). **Bottom:** The "reconstruction" $p_\theta(x|z)$ when z is sampled from $q_\phi(z|x = 1)$ (green) and $q_\phi(z|x = -1)$ (red). Also plotted is $p_\theta(x|z)$ when z is sampled from the true prior $p(z)$ (purple). When the dataset consists of only two data points, ELBO (**left**) will push the density in latent space Z away from 0, while InfoVAE (**right**) does not suffer from this problem.

The $\mathcal{L}_{InfoVAE}$ is given by

$$-\lambda D_{KL}(q_\phi(z)||p_\theta(z)) - \mathbb{E}_{q_{\phi(z)}} D_{KL}(q_\phi(x|z)||p_\theta(x|z)) + \alpha I_{q_\phi}(x; z)$$

where $I_\phi(x; z)$ is the mutual information between x and z under the joint distribution $q_\phi(x, z)$. VAE $\mathcal{L}_{ELBO}$ is recovered for $\alpha = 0$ and $\lambda = 1$. β-VAE loss is recovered for $\alpha = 1 - \lambda$.

[2] `https://arxiv.org/pdf/1706.02262.pdf`

Information Bottleneck for DNN (Tishby et al. 15'[3])

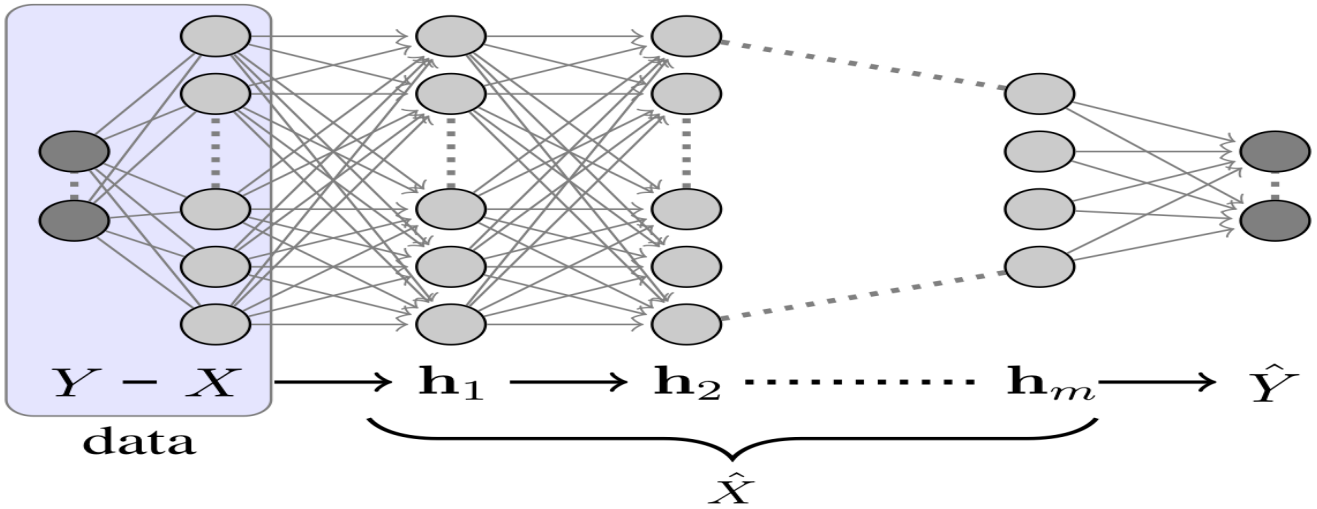


Fig. 1. An example of a feedforward DNN with m hidden layers, an input layer X and an output layer $\hat{Y}$. The desired output, Y, is observed only during the training phase through a finite sample of the joint distribution, $p(X, Y)$, and is used for learning the connectivity matrices between consecutive layers. After training, the network receives an input X, and successively processes it through the layers, which form a Markov chain, to the predicted output $\hat{Y}$. $I(Y;\hat{Y})/I(X;Y)$ quantifies how much of the relevant information is captured by the network.

$$I(Y;X) \geq I(Y;h_1) \geq \cdots \geq I(Y;h_m) \geq I(Y;\hat{Y})$$

The mutual information $I(Y;\hat{Y})$ is maximized by maximizing $I(Y;h_i)$ for each layer.

[3] https://arxiv.org/pdf/1503.02406.pdf

Information Bottleneck: compression (Tishby et al. 15'[4])

Consider the data, X and desired output Y. The *minimal sufficient statistic* of X to represent Y is denoted by $\hat{X}$ and is the simplest representation of X to describe Y.

Considering the map $Y \to X \to \hat{X}$ from the desired output to X and then its simplest representation $\hat{X}$ we have $I(Y;X) \geq I(Y;\hat{X})$ with the gap minimized by minimizing $I(X;\hat{X})$. The optimal $\hat{X}$ follows from minimizing

$$I(X;\hat{X}) + \beta \mathbb{E} D_{KL}(p(y|x)||p(y|\hat{x})) =: I(X;\hat{X}) + \beta D_{IB}$$

We can then view the layers as learning representations in terms of the information retained at a layer

$$I(h_{i-1};h_i) + \beta I(Y;h_{i-1}:h_i).$$

[4] https://arxiv.org/pdf/1503.02406.pdf

Information Bottleneck: finite data (Tishby et al. 15'[5])

Consider the data X and its sufficient statistic $\hat{X}$ to have some cardinality in terms of their representation, e.g. a measure of the support of the data manifold, and let $\hat{I}$ be the empirical estimate of the mutual information based on the finite sample distribution $\hat{p}(x, y)$ from n samples, then the *generalization bound* is given by

$$I(\hat{X}; Y) \leq \hat{I}(\hat{X}; Y) + \mathcal{O}\left(\frac{|\hat{X}| \cdot |Y|}{\sqrt{n}}\right)$$

and

$$R := I(X; \hat{X}) \leq \hat{I}(X; \hat{X}) + \mathcal{O}\left(\frac{|\hat{X}|}{\sqrt{n}}\right)$$

which illustrate the dependence on complexity is given in terms of the sufficient statistic $\hat{X}$ complexity defined by Y rather than the ambient dimension of X.

[5] https://arxiv.org/pdf/1503.02406.pdf

Information Bottleneck: finite data (Tishby et al. 15'[6])

Let $D_N = I(X;Y|\hat{Y})$ and $R_N = I(X;\hat{Y})$ be the information bottleneck distortion and representation for the network. Then the *generalization gap* is $\Delta G := D_N - D_{IB}(n)$ and the complexity gap is $\Delta C := R_N - R(n)$.

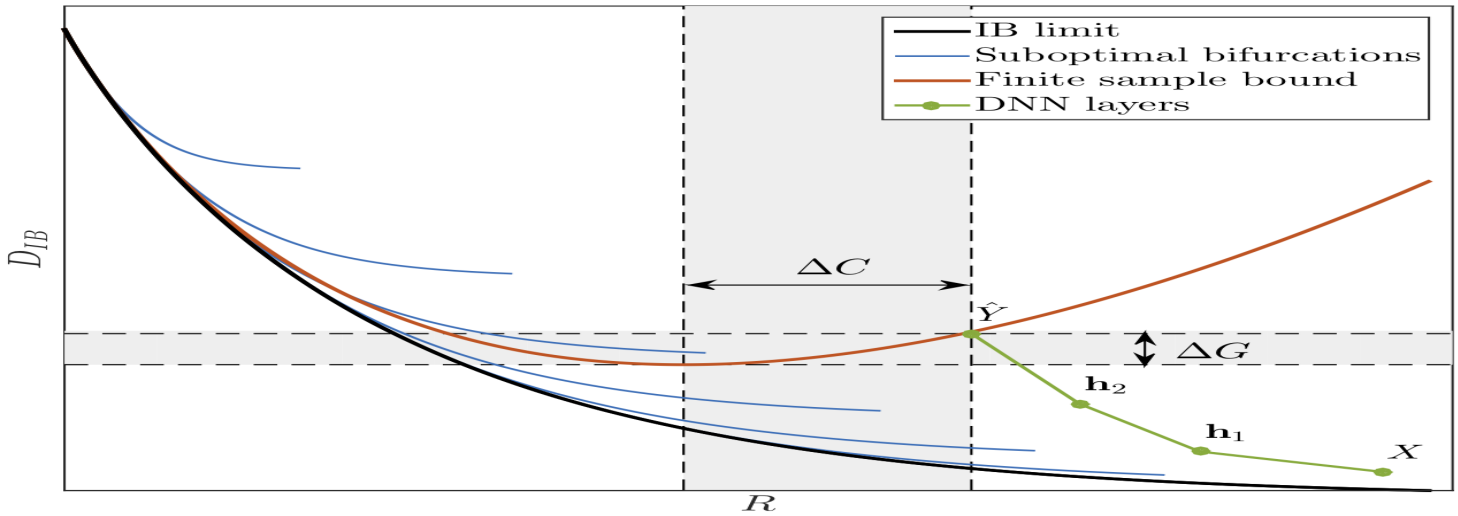


Fig. 2. A qualitative information plane, with a hypothesized path of the layers in a typical DNN (green line) on the training data. The black line is the optimal achievable IB limit, and the blue lines are sub-optimal IB bifurcations, obtained by forcing the cardinality of $\hat{X}$ or remaining in the same representation. The red line corresponds to the upper bound on the *out-of-sample* IB distortion (mutual information on Y), when training from a finite sample. While the training distortion may be very low (the green points) the actual distortion can be as high as the red bound. This is the reason why one would like to shift the green DNN layers closer to the optimal curve to obtain lower complexity and better generalization. Another interesting consequence is that getting closer to the optimal limit requires stochastic mapping between the layers.

[6] https://arxiv.org/pdf/1503.02406.pdf

Information Bottleneck plane (Schwartz-Ziv et al. 17'[7])

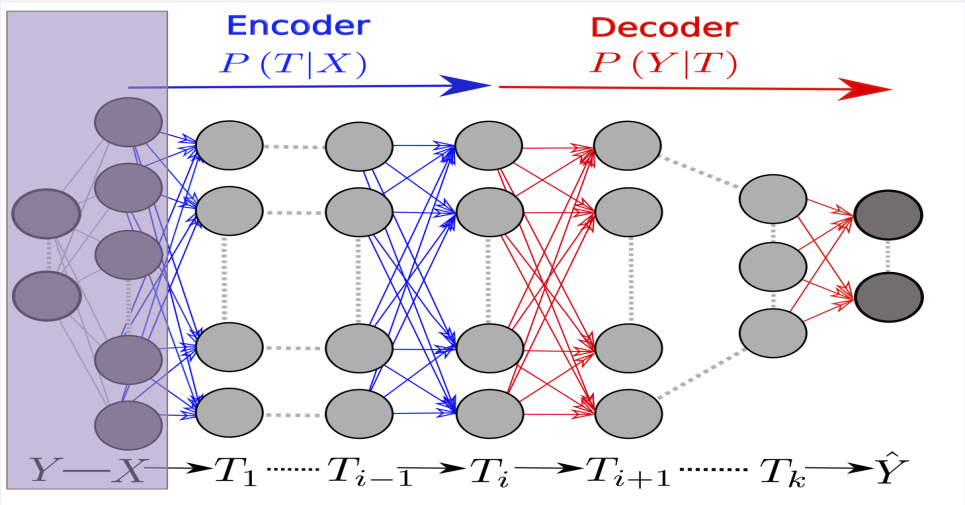


Figure 1: The DNN layers form a Markov chain of successive internal representations of the input layer X. Any representation of the input, T, is defined through an encoder, $P(T|X)$, and a decoder $P(\hat{Y}|T)$, and can be quantified by its *information plane* coordinates: $I_X = I(X;T)$ and $I_Y = I(T;Y)$. The Information Bottleneck bound characterizes the optimal representations, which maximally compress the input X, for a given mutual information on the desired output Y. After training, the network receives an input X, and successively processes it through the layers, which form a Markov chain, to the predicted output $\hat{Y}$. $I(Y;\hat{Y})/I(X;Y)$ quantifies how much of the relevant information is captured by the network.

note, change of notation of hidden layers from h_i to T_i.

[7] https://arxiv.org/pdf/1703.00810.pdf

Information Bottleneck plane (Schwartz-Ziv et al. 17'[8])

Consider the 2d plane expressing the quantities $I(X, T_j)$ vs. $I(T_j, Y)$ measuring how much layer j correlates the information between a layer and the output as a function of the information between the input and the same layer.

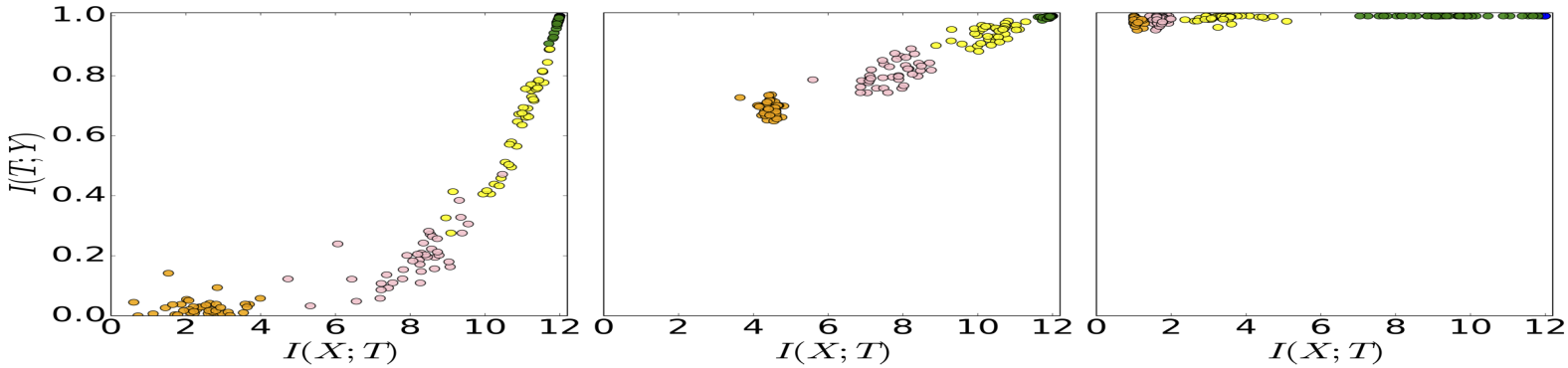


Figure 2: Snapshots of layers (different colors) of 50 randomized networks during the SGD optimization process in the *information plane* (in bits): **left** - with the initial weights; **center** - at 400 epochs; **right** - after 9000 epochs. The reader is encouraged to view the full videos of this optimization process in the *information plane* at *https://goo.gl/rygyIT* and *https://goo.gl/DQWuDD*.

$$I(X;Y) \geq I(T_1;Y) \geq \cdots \geq I(T_K;Y) \geq I(\hat{Y};Y)$$

$$H(X) \geq I(X;T_1) \geq \cdots \geq I(X;T_k) \geq I(X;\hat{Y})$$

[8] https://arxiv.org/pdf/1703.00810.pdf

Information Bottleneck plane (Schwartz-Ziv et al. 17'[9])

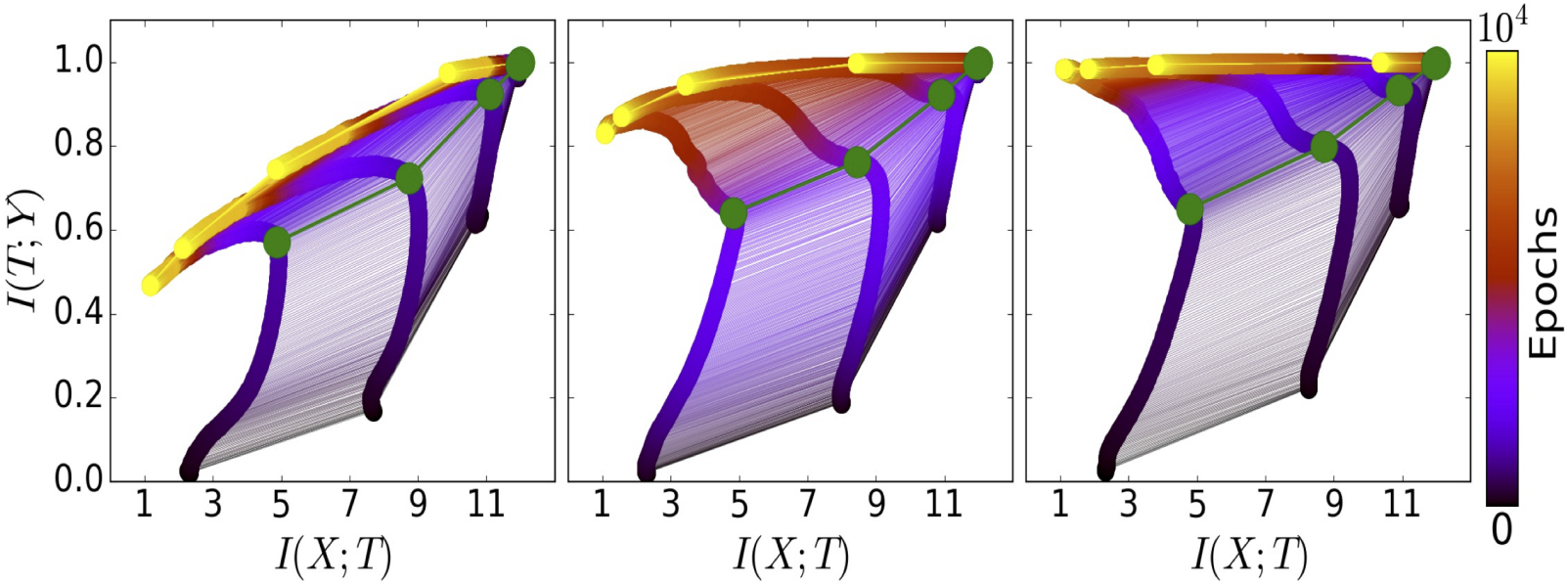


Figure 3: The evolution of the layers with the training epochs in the information plane, for different training samples. On the left - 5% of the data, middle - 45% of the data, and right - 85% of the data. The colors indicate the number of training epochs with Stochastic Gradient Descent from 0 to 10000. The network architecture was fully connected layers, with widths: input=12-10-8-6-4-2-1=output. The examples were generated by the spherical symmetric rule described in the text. The green paths correspond to the SGD drift-diffusion phase transition - grey line on Figure 4

[9] `https://arxiv.org/pdf/1703.00810.pdf`

Information Bottleneck plane (Schwartz-Ziv et al. 17'[10])

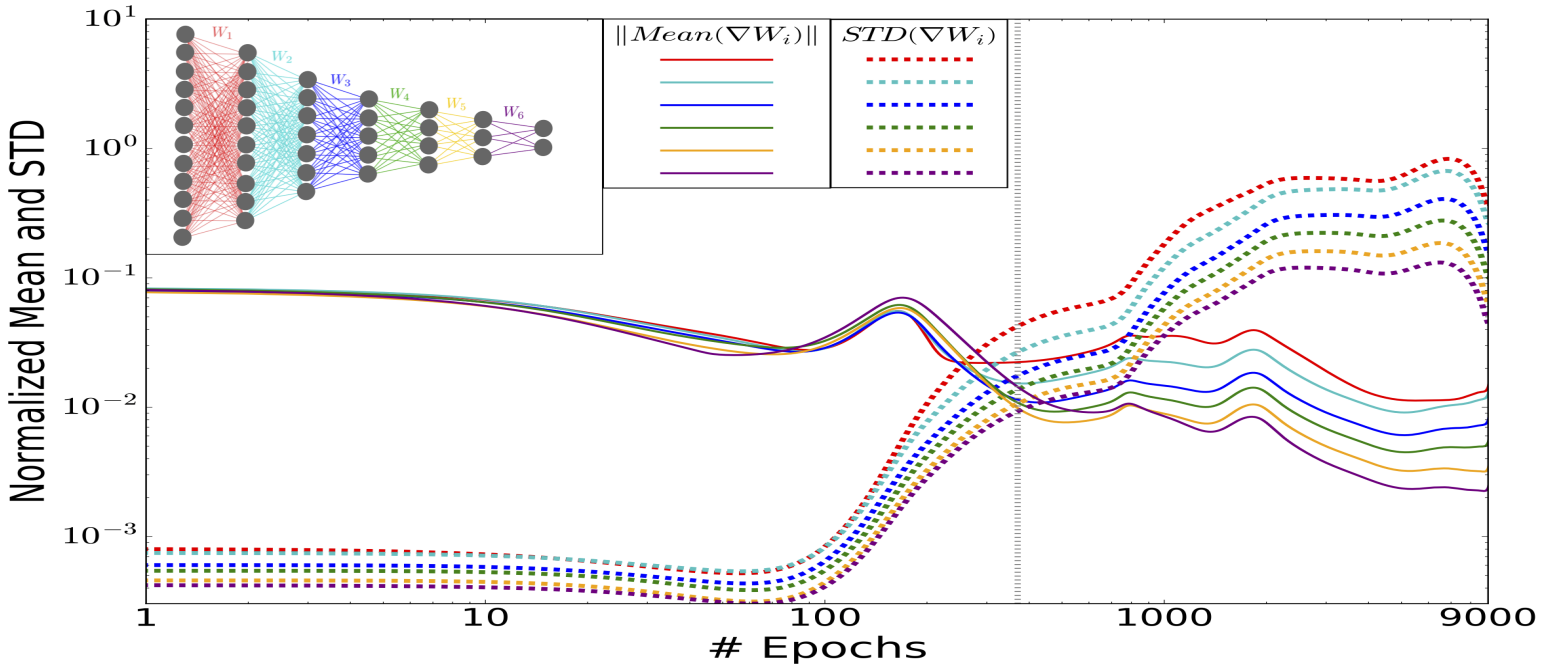


Figure 4: **The layers' Stochastic Gradients distributions during the optimization process.** The norm of the means and standard deviations of the weights gradients for each layer, as function of the number of training epochs (in log-log scale). The values are normalized by the L2 norms of the weights for each layer, which significantly increase during the optimization. The grey line ($\sim$ 350 epochs) marks the transition between the first phase, with large gradient means and small variance (*drift*, high gradient SNR), and the second phase, with large fluctuations and small means (*diffusion*, low SNR). Note that the gradients log (SNR) (the log differences between the mean and the STD lines) approach a constant for all the layers, reflecting the convergence of the network to a configuration with constant flow of relevant information through the layers!

[10] https://arxiv.org/pdf/1703.00810.pdf

Information Bottleneck plane (Schwartz-Ziv et al. 17'[11])

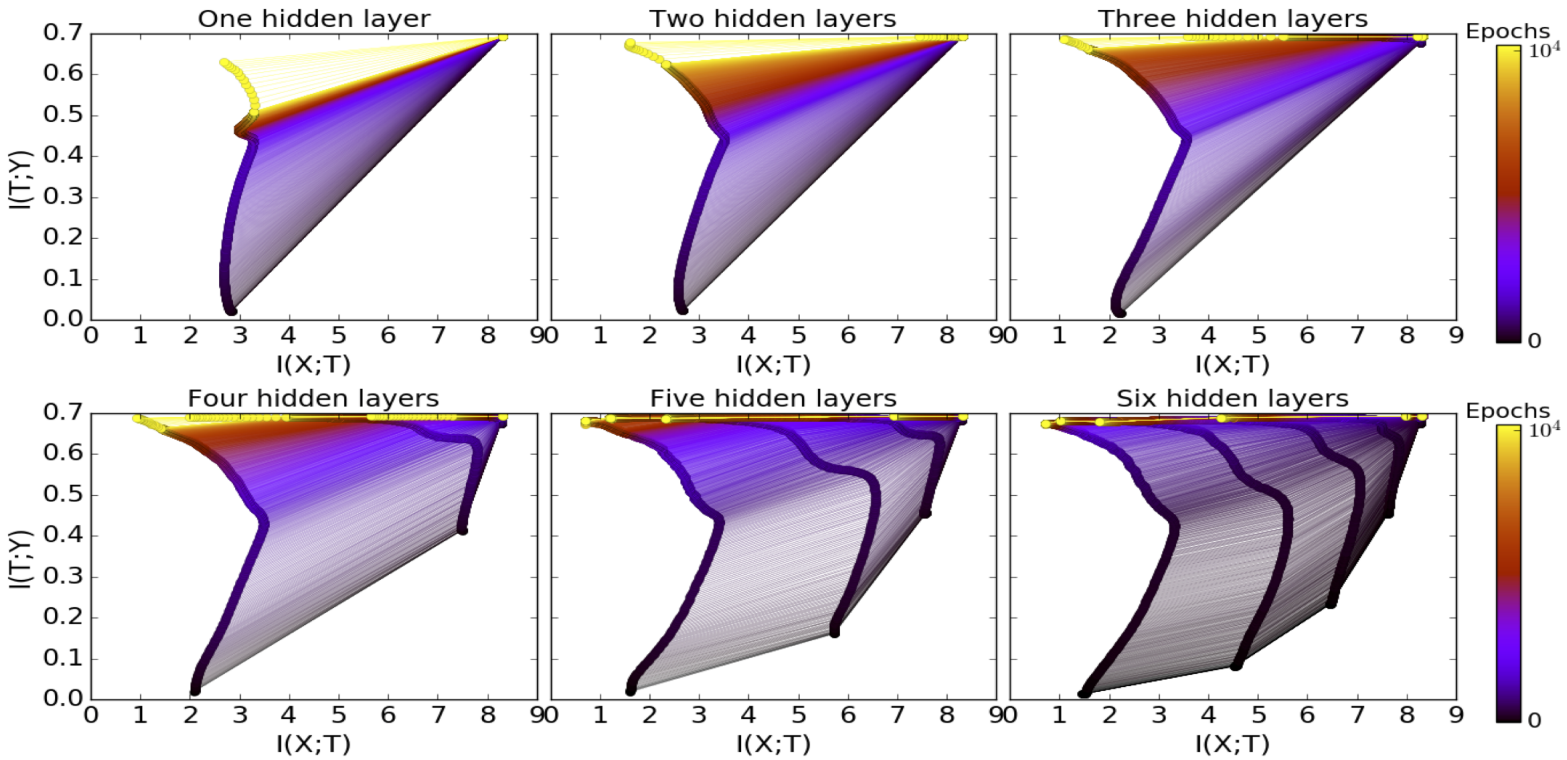


Figure 5: **The layers information paths during the SGD optimization for different architectures.** Each panel is the *information plane* for a network with a different number of hidden layers. The width of the hidden layers start with 12, and each additional layer has 2 fewer neurons. The final layer with 2 neurons is shown in all panels. The line colors correspond to the number of training epochs.

[11] https://arxiv.org/pdf/1703.00810.pdf

Information Bottleneck plane (Schwartz-Ziv et al. 17'[12])

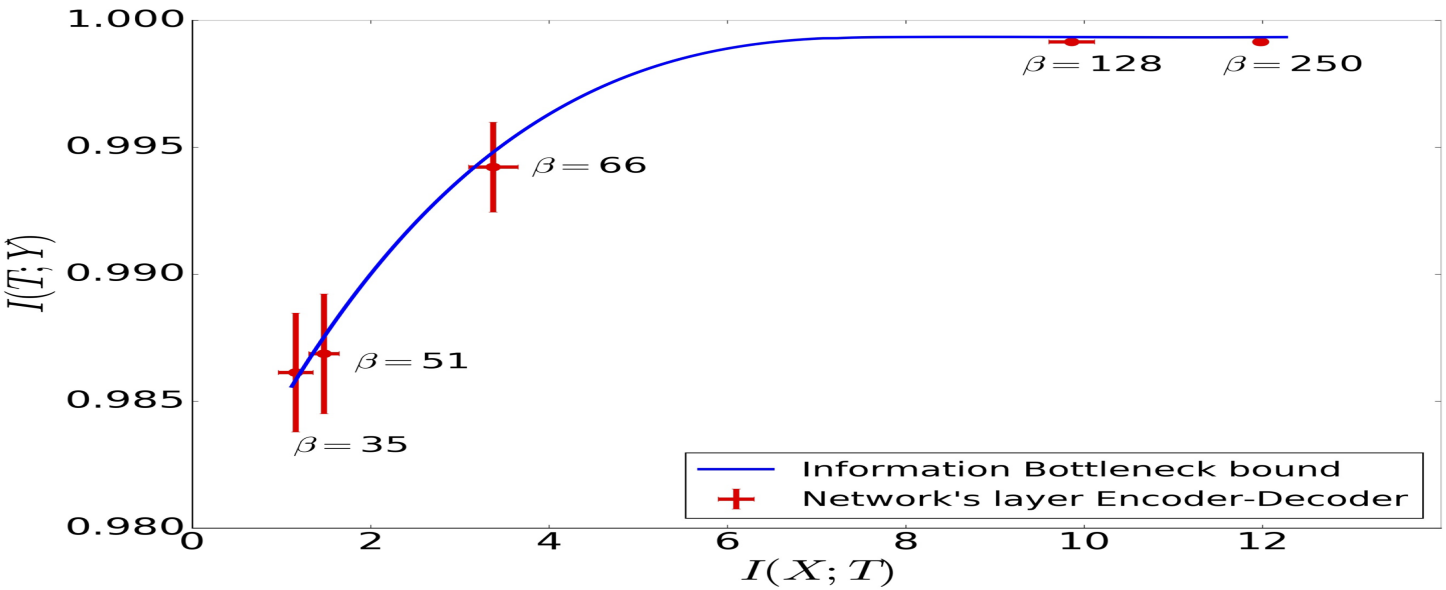


Figure 6: **The DNN layers converge to fixed-points of the IB equations**. The error bars represent standard error measures with N=50. In each line there are 5 points for the different layers. For each point, β is the optimal value that was found for the corresponding layer.

The information bottleneck tradeoff is given by minimizing

$$\min_{p(t|x),p(y|t),p(t)} I(X;T) - \beta I(T;Y)$$

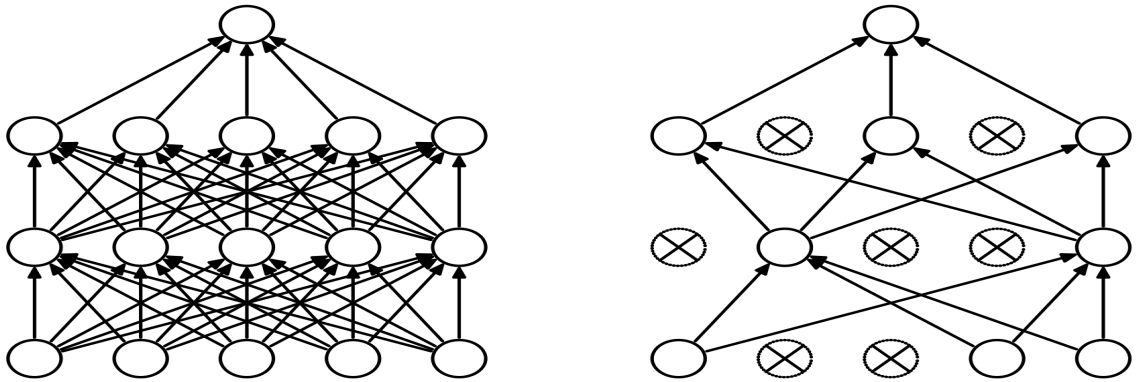
where β balances the tradeoff of representing X and Y.
Plotted in the figure is per layer the minimizer of

$$\mathbb{E}_x D_{KL}(p(t_i|x)||p_\beta^{IB}(t_i|x))$$

[12] [illegible]

Dropout (Srivastava et al. 14'[13])

Dropout is a method by which, during training, the activations are set to zero with some probability. Note, dropout is only used in the training phase, not in testing.



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

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

Dropout has a number of valuable consequences, ranging from improving sparsity in the net, improving testing accuracy, avoiding overfitting, and can be used to evaluate uncertainty in a deep net.

[13] `http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf`

Dropout: test error (Srivastava et al. 14'[14])

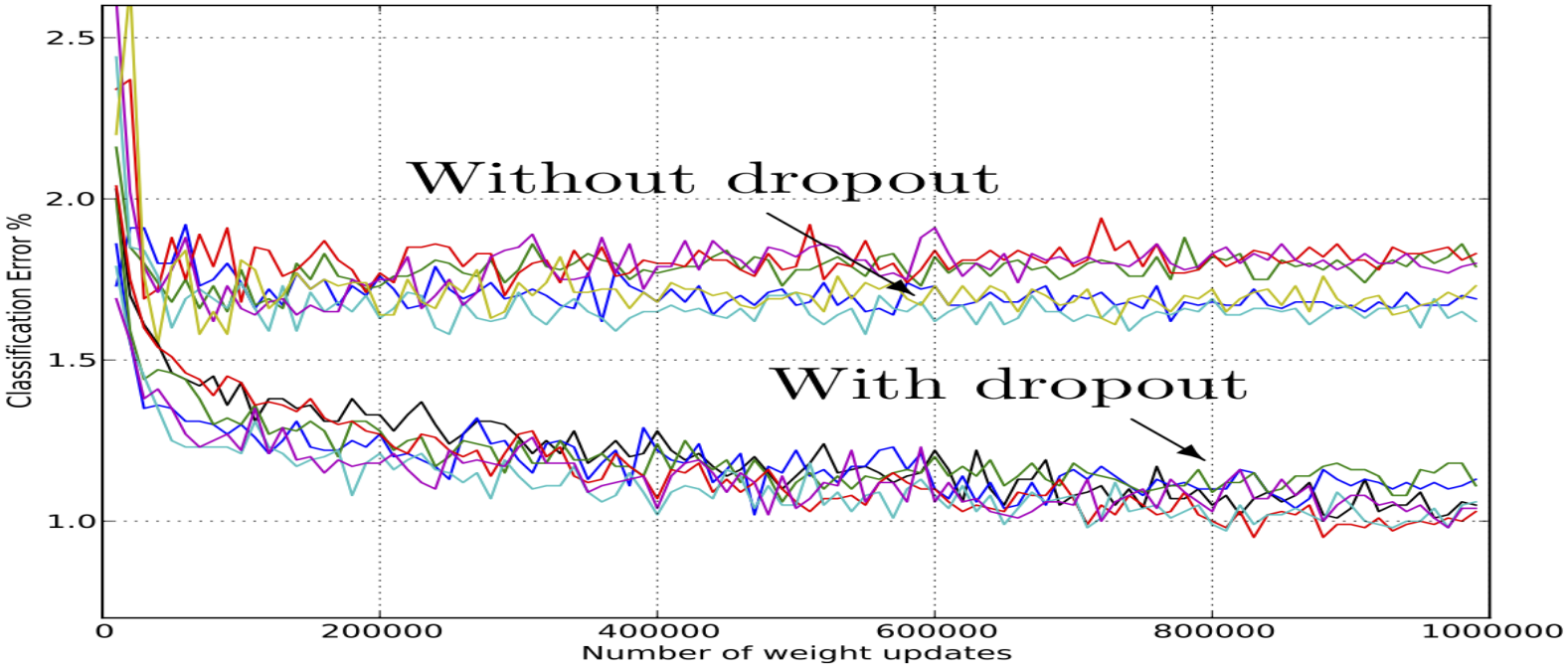


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

[14] http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

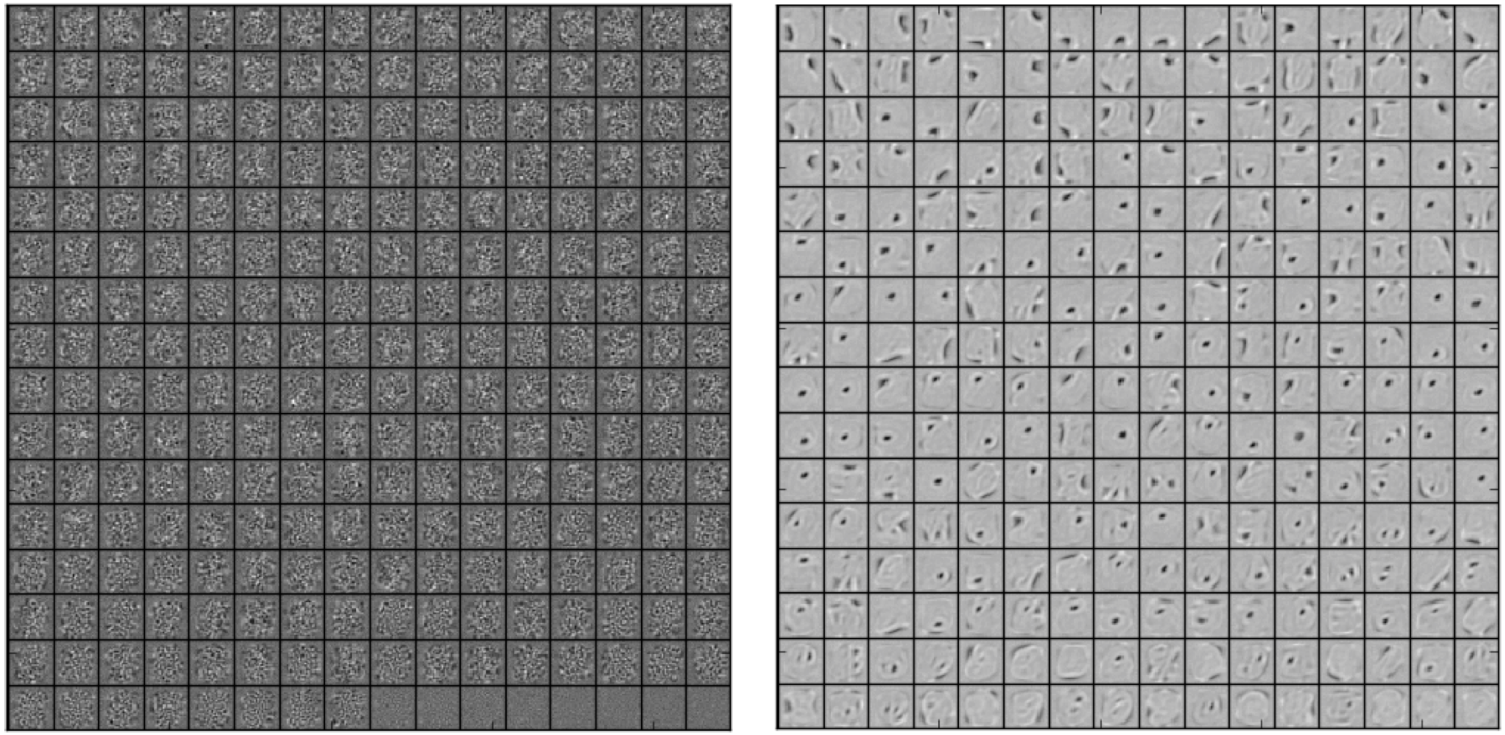
Dropout: street house numbers (Srivastava et al. 14'[15])

Method	Error %
Binary Features (WDCH) (Netzer et al., 2011)	36.7
HOG (Netzer et al., 2011)	15.0
Stacked Sparse Autoencoders (Netzer et al., 2011)	10.3
KMeans (Netzer et al., 2011)	9.4
Multi-stage Conv Net with average pooling (Sermanet et al., 2012)	9.06
Multi-stage Conv Net + L2 pooling (Sermanet et al., 2012)	5.36
Multi-stage Conv Net + L4 pooling + padding (Sermanet et al., 2012)	4.90
Conv Net + max-pooling	3.95
Conv Net + max pooling + dropout in fully connected layers	3.02
Conv Net + stochastic pooling (Zeiler and Fergus, 2013)	2.80
Conv Net + max pooling + dropout in all layers	2.55
Conv Net + maxout (Goodfellow et al., 2013)	**2.47**
Human Performance	2.0

Table 3: Results on the Street View House Numbers data set.

[15] `http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf`

Dropout: sparse features (Srivastava et al. 14'[16])



(a) Without dropout

(b) Dropout with $p = 0.5$.

Figure 7: Features learned on MNIST with one hidden layer autoencoders having 256 rectified linear units.

[16] http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf