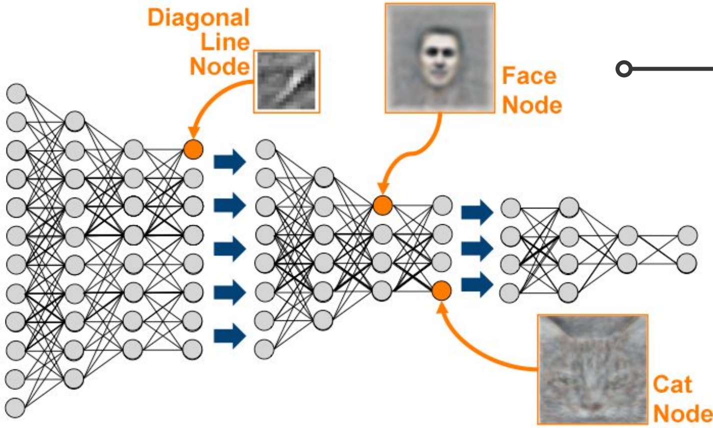
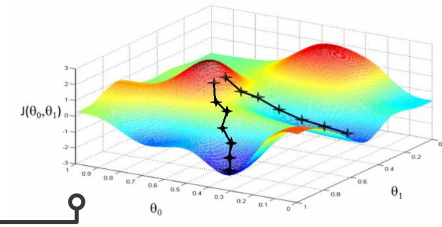
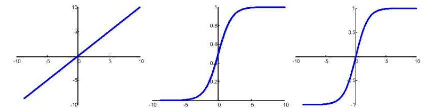


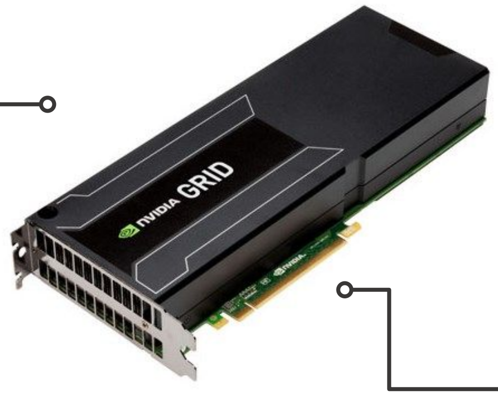


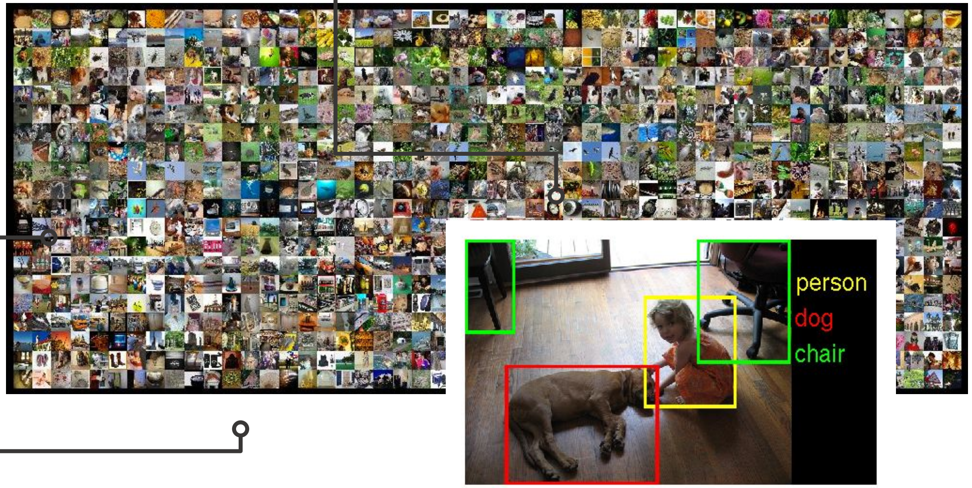


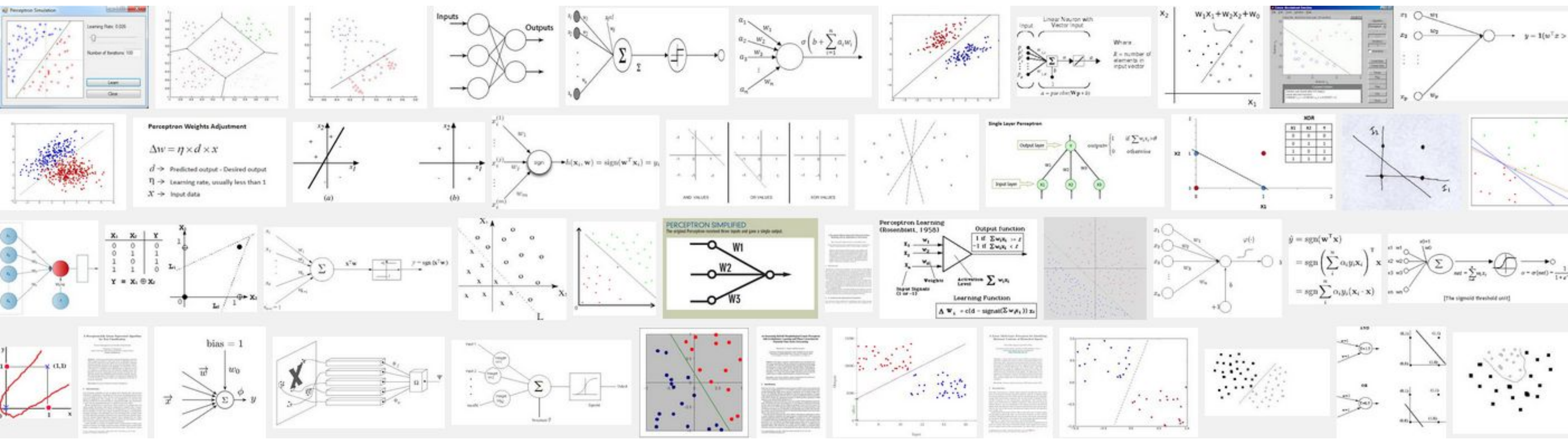
Métodos Actuales de Machine Learning

Neural Networks

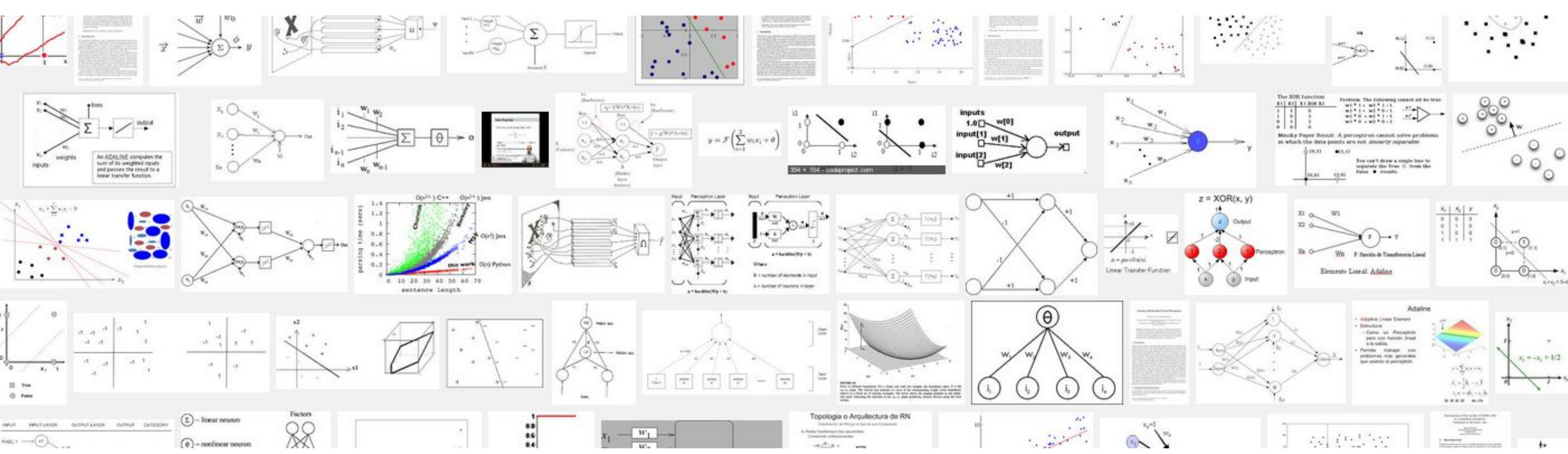
http://www.cifasis-conicet.gov.ar/granitto/RIO2016/



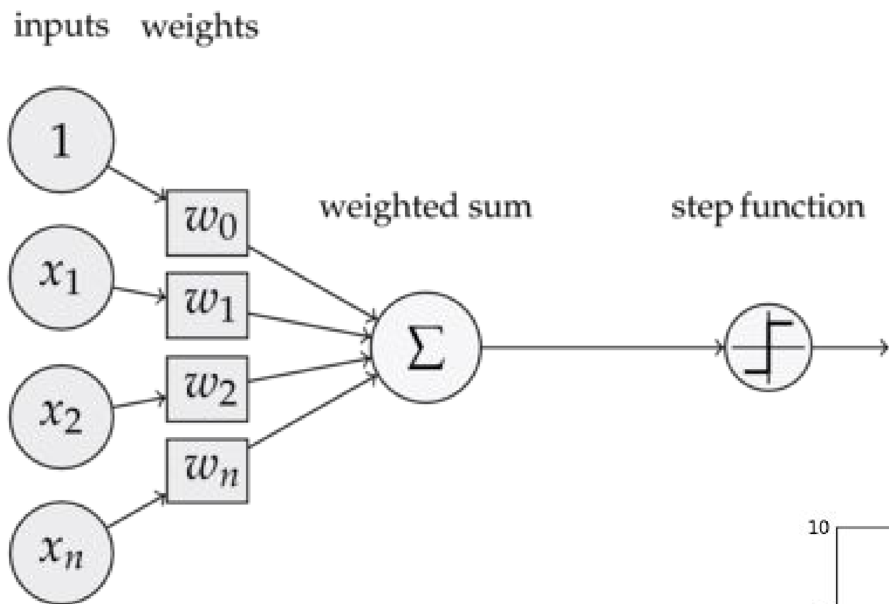




Linear Perceptron

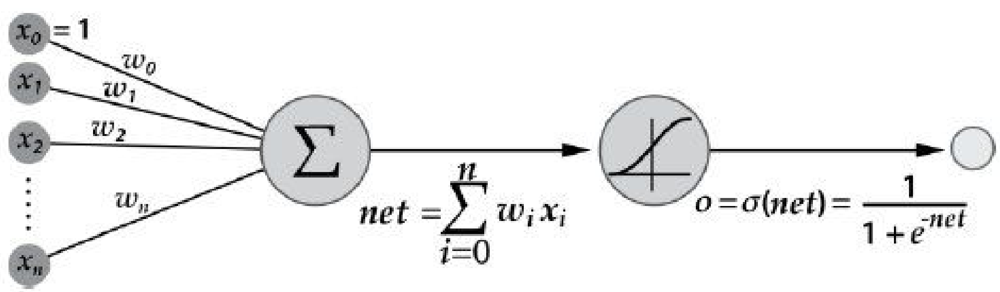


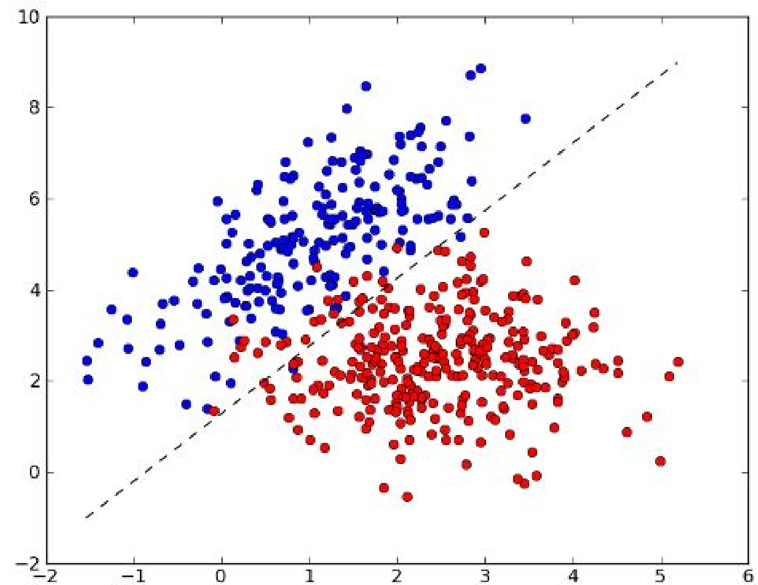
Linear Perceptron



1957

$$g_0(x) = \sigma(x^T W^{(0)} + b^{(0)})$$





Softmax Regression

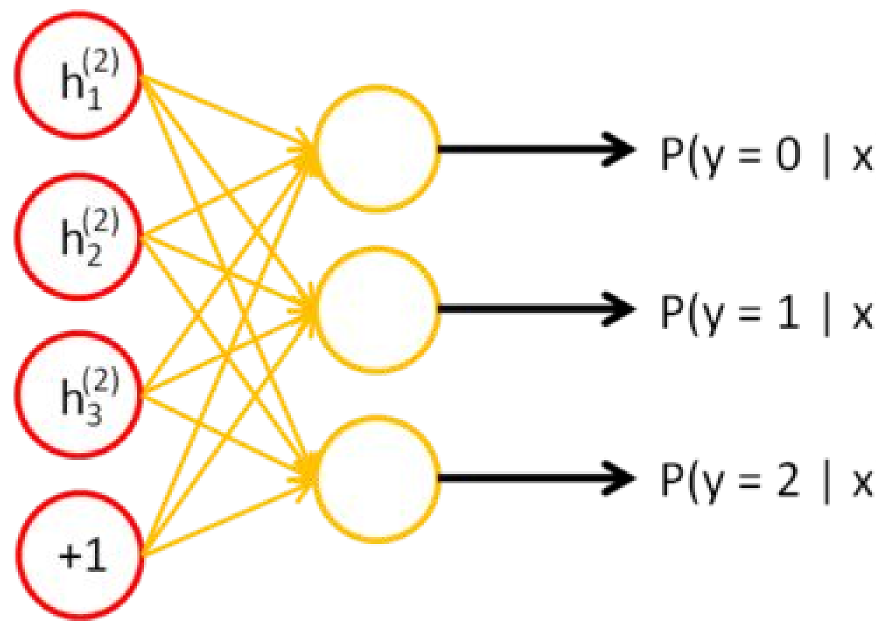
$$\log \overbrace{p(y|x)}^{\text{class probability vector}} = \overbrace{\underbrace{x^T W + b}_{\text{affine transform.}} + \underbrace{c(x)}_{\text{normalization}}}^{\text{parametric model}}$$

output
(multiple categories)

input

$$p(y|x) = \frac{\exp(x^T W + b)}{\sum_i \exp(x^T W + b)_i}$$

$$= \mathrm{softmax}(x^T W + b)$$



Softmax Regression Training

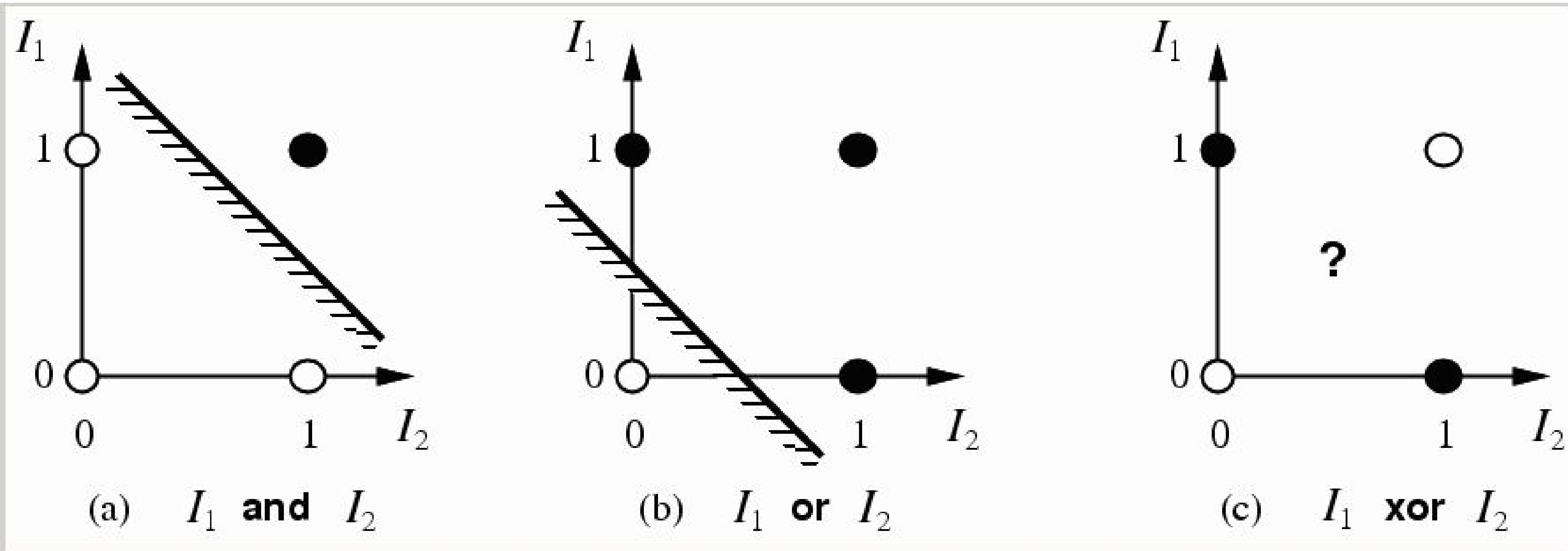
$$J(\mathcal{D}, W, b) = \prod_{x,y \in \mathcal{D}} p(y|x)$$ Maximum Likelihood Estimation (MLE)

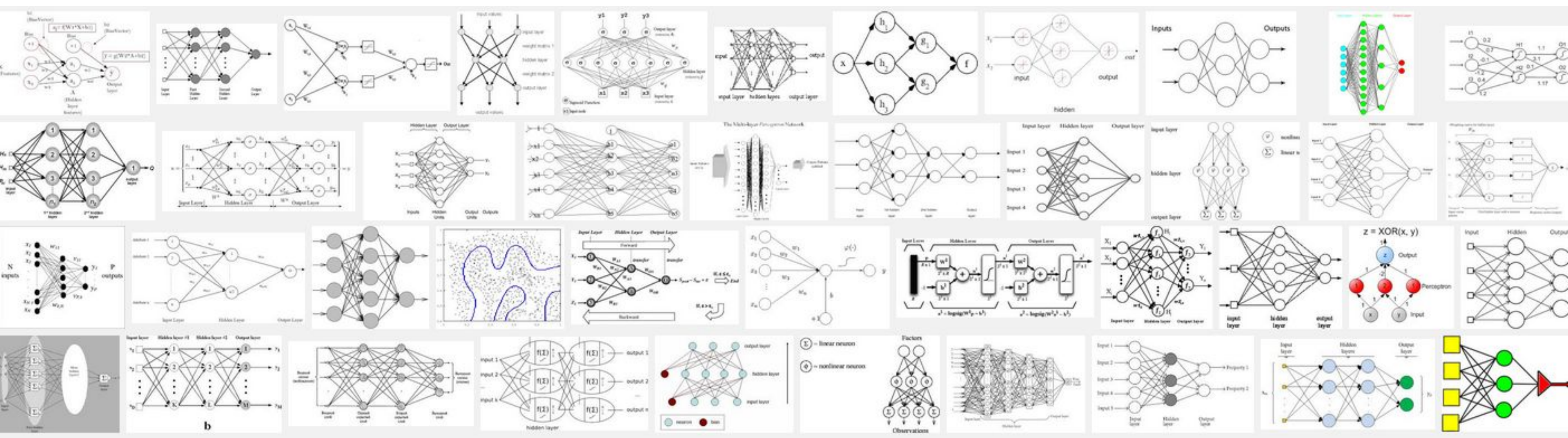
training dataset

$$J(\mathcal{D}, W, b) = \sum_{x,y \in \mathcal{D}} \log p(y|x)$$ Log-Likelihood

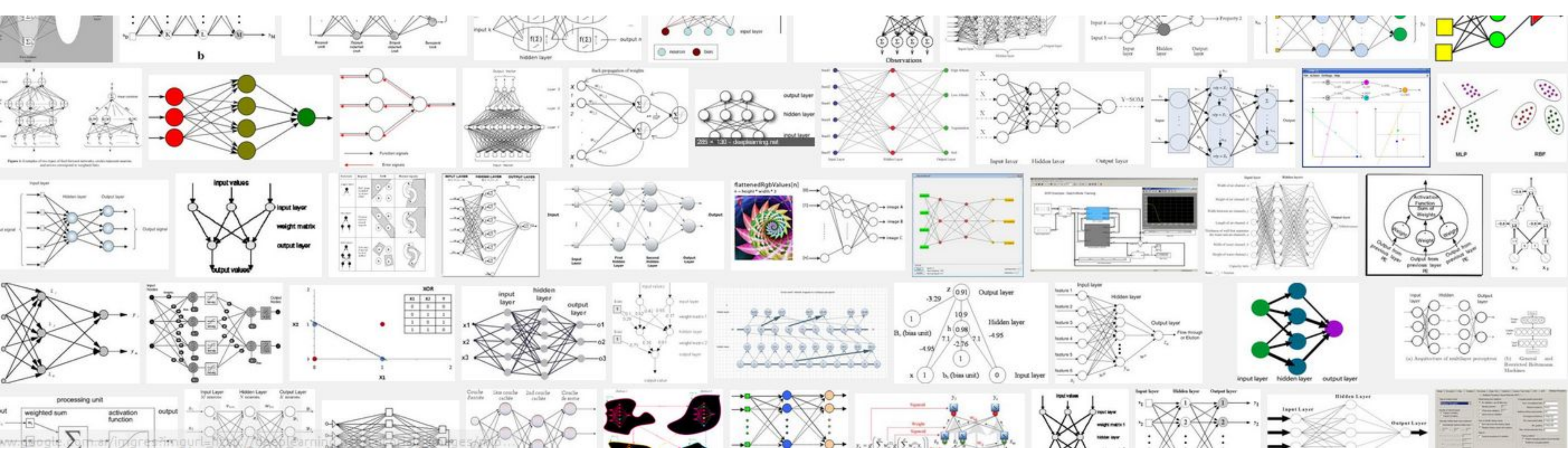
$$NLL(\theta, \mathcal{D}) = -\sum_{i=0}^{|\mathcal{D}|} \log P(Y = y^{(i)}|x^{(i)}, \theta)$$ Negative Log-Likelihood

Linear Perceptron





Multilayer Perceptrons (MLP)



Multilayer Perceptrons (MLP)

$$J(\mathcal{D}, W, b) = \sum_{x,y \in \mathcal{D}} \log p(y|x)$$

Log-Likelihood

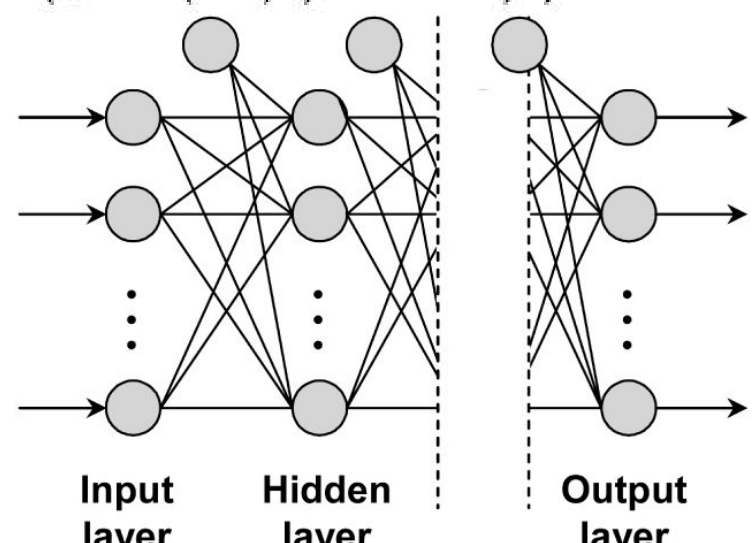
$$\log p(y|x) = \cancel{x^T W + b + c(x)}$$

Single layer softmax

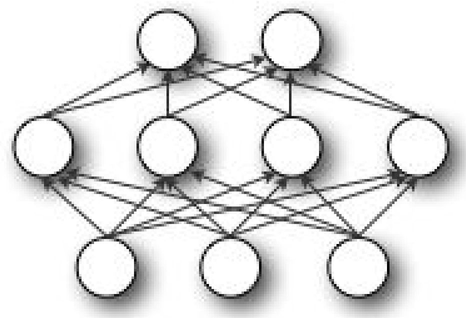
MLP output

$$\underbrace{f(x)}_{\text{MLP output}} = g_L(g_{L-1}(\cdots g_2(g_1(x)) \ldots))$$

$$g_\ell(x) = \sigma(x^T W^{(\ell)} + b^{(\ell)})$$



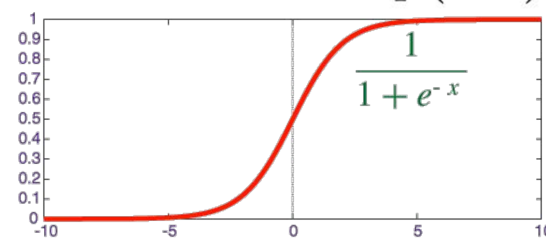
Multilayer Perceptrons (MLP)



$$g_2(g_1) = \text{softmax}(g_1^T W^{(2)} + b^{(2)})$$

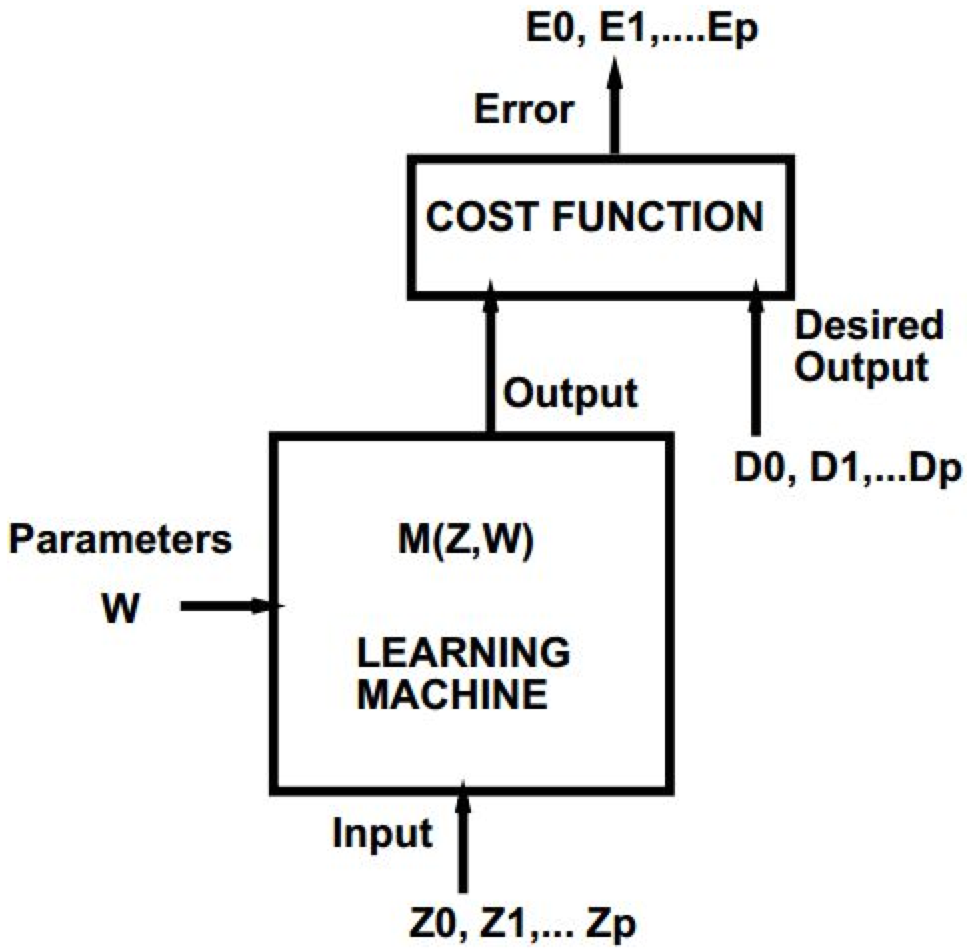
$$g_1(x) = \sigma(x^T W^{(1)} + b^{(1)})$$

$$\sigma(z) = \frac{1}{1+exp(-z)}$$



$$f(x) = \text{softmax}\left(\sigma(x^T W^{(1)} + b^{(1)})^T W^{(2)} + b^{(2)}\right)$$

Gradient Based Learning Machine



$$E^p = \frac{1}{2}(D^p - M(Z^p, W))^2$$

$$E_{train} = \frac{1}{P}\sum_{p=1} E^p$$

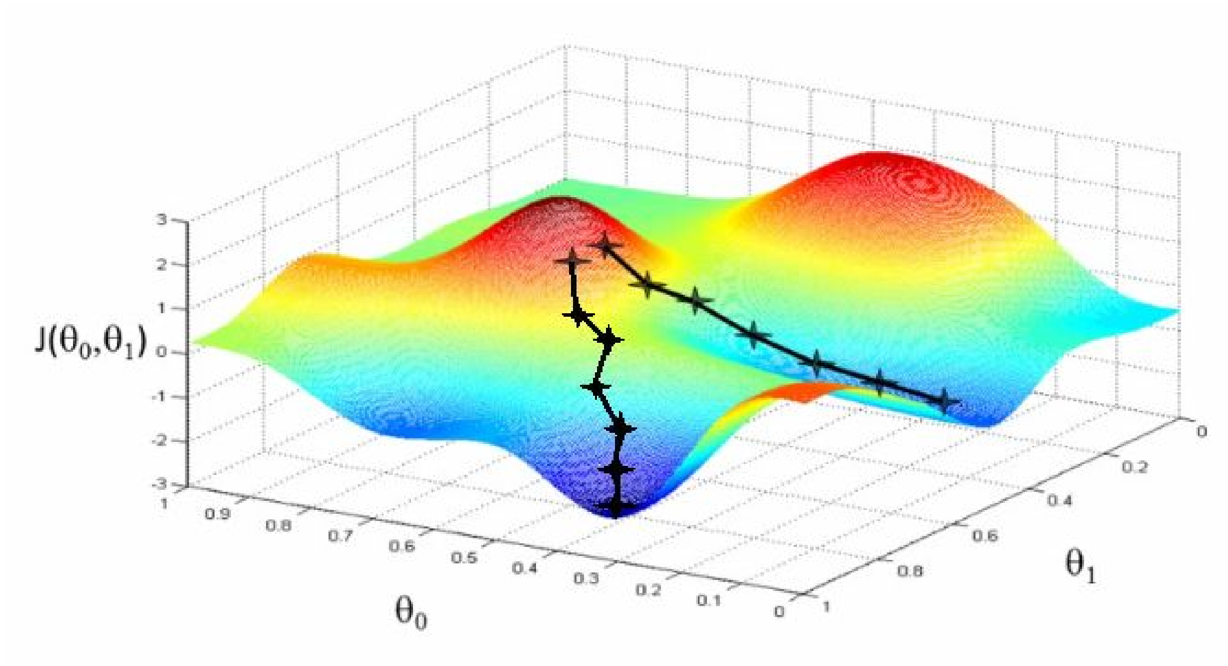
$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W}$$

LeCun, Yann A et al. "Efficient backprop." *Neural networks: Tricks of the trade* (2012): 9-48.

Gradient Descent

Algorithm 1 GRADIENT DESCENT

while True **do**
  loss = f(params)
  d_loss_wrt_params = ... ▷ compute gradient
  params -= learning_rate * d_loss_wrt_params
  if stopping condition is met **then return** params
  end if
end while



Stochastic Gradient Descent

Algorithm 1 GRADIENT DESCENT

while True **do**
  loss = f(params)
  d_loss_wrt_params = ... ▷ compute gradient
  params -= learning_rate * d_loss_wrt_params
  if stopping condition is met **then return** params
  end if
end while

Algorithm 2 STOCHASTIC GRADIENT DESCENT

for $(x_i, y_i) \in \mathcal{D}_{train}$ **do**
  ▷ imagine an infinite generator that may repeat
  ▷ examples (if there is only a finite training set)
  loss = f(params, x_i, y_i)
  d_loss_wrt_params = ... ▷ compute gradient
  params $-$ = learning_rate * d_loss_wrt_params
  if stopping condition is met **then return** params
  end if
end for

Mini-batch Gradient Descent

Algorithm 2 STOCHASTIC GRADIENT DESCENT

for $(x_i, y_i) \in \mathcal{D}_{train}$ **do**
▷ imagine an infinite generator that may repeat
▷ examples (if there is only a finite training set)
    loss = f(params, x_i, y_i)
    d_loss_wrt_params = ... ▷ compute gradient
    params − = learning_rate * d_loss_wrt_params
    if stopping condition is met **then return** params
    end if
end for

Algorithm 3 MINIBATCH SGD

for (x_batch,y_batch) ∈ train_batches **do**
▷ imagine an infinite generator
▷ that may repeat examples
    loss = f(params, x_batch, y_batch)
    d_loss_wrt_params = ... ▷ compute gradient
    params − = learning_rate * d_loss_wrt_params
    if stopping condition is met **then return** params
    end if
end for

Hyperparameters: Learning Rate

$$W_i \leftarrow W_i - \eta \frac{\partial \mathrm{E}(W_i)}{\partial W_i}$$

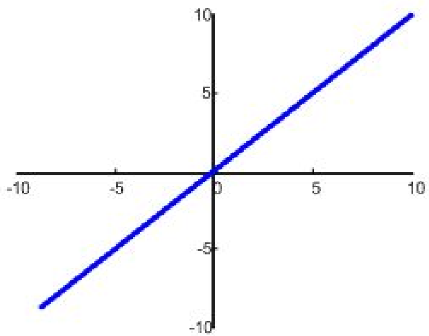
learning rate

- constant learning rate (simplest solution)
- logarithmic grid search (10^{-1}, 10^{-2}, ...)
- decreasing learning rate over time:

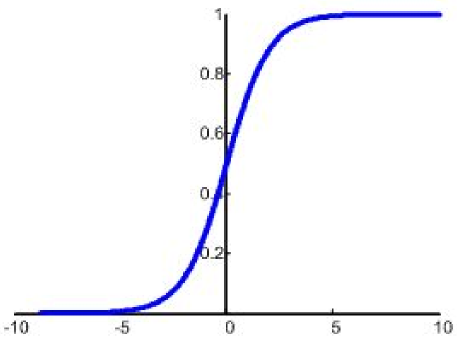
$$\eta_t = \frac{\eta_0}{1 + at}$$

For adaptive learning rate see: LeCun, Yann A et al. "Efficient backprop." *Neural networks: Tricks of the trade* (2012): 9-48.

Hyperparameters: Nonlinearity

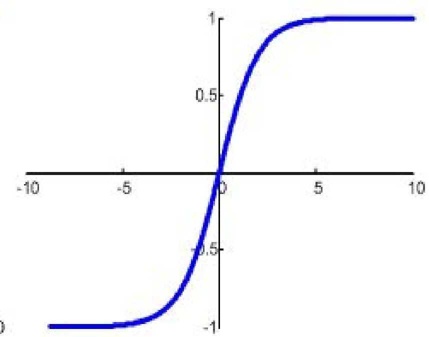


linear



logistic

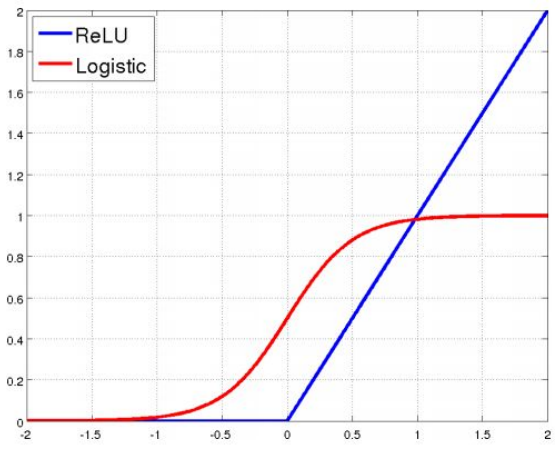
$$f(u) = 1/(1 + \exp(-u))$$



tanh

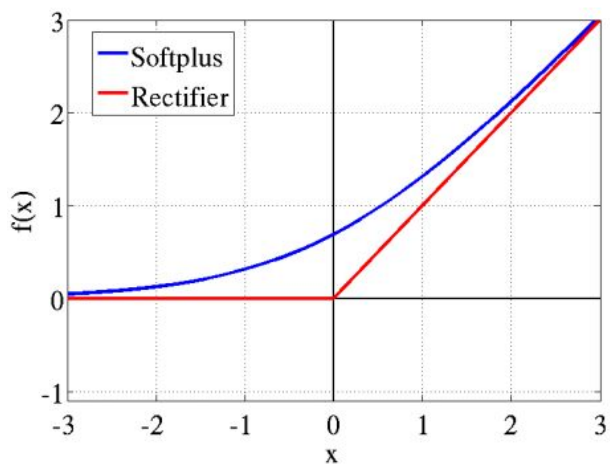
ReLU

$$f(u) = \max(0, u)$$

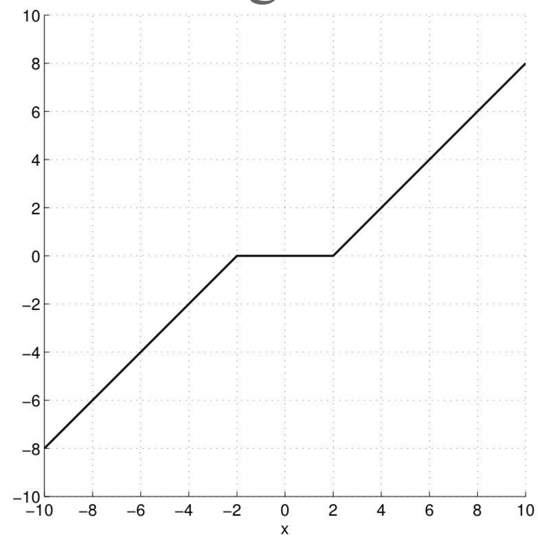


Softplus

$$softplus(x) = log(1+e^x)$$



Shrinkage



Hyperparameters: momentum

$$\Delta\theta_i(t) = v_i(t) = \alpha v_i(t-1) - \epsilon \frac{dE}{d\theta_i}(t)$$

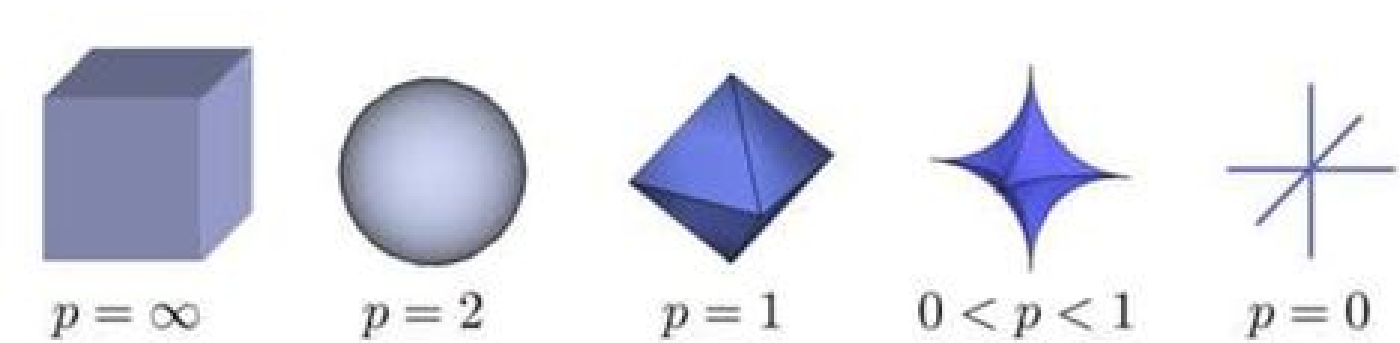
momentum learning rate

Hinton, Geoffrey E. "A practical guide to training restricted boltzmann machines." *Neural Networks: Tricks of the Trade* (2012): 599-619.

Regularization

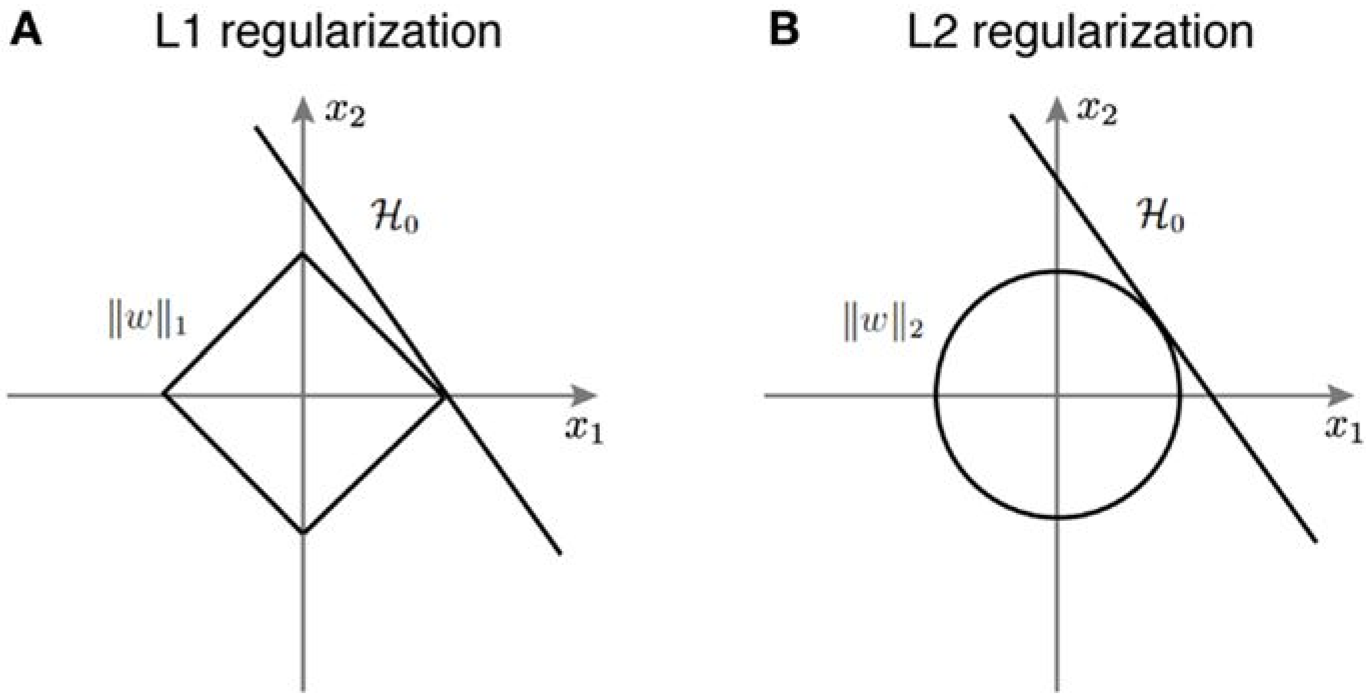
$$E(\theta, \mathcal{D}) = \underbrace{NLL(\theta, \mathcal{D})}_{\text{cost function}} + \underbrace{\lambda \|\theta\|_p^p}_{\text{regularization term}}$$

p-norm of θ $\quad \|\theta\|_p = \left(\sum_{j=0} |\theta_j|^p \right)^{\frac{1}{p}}$



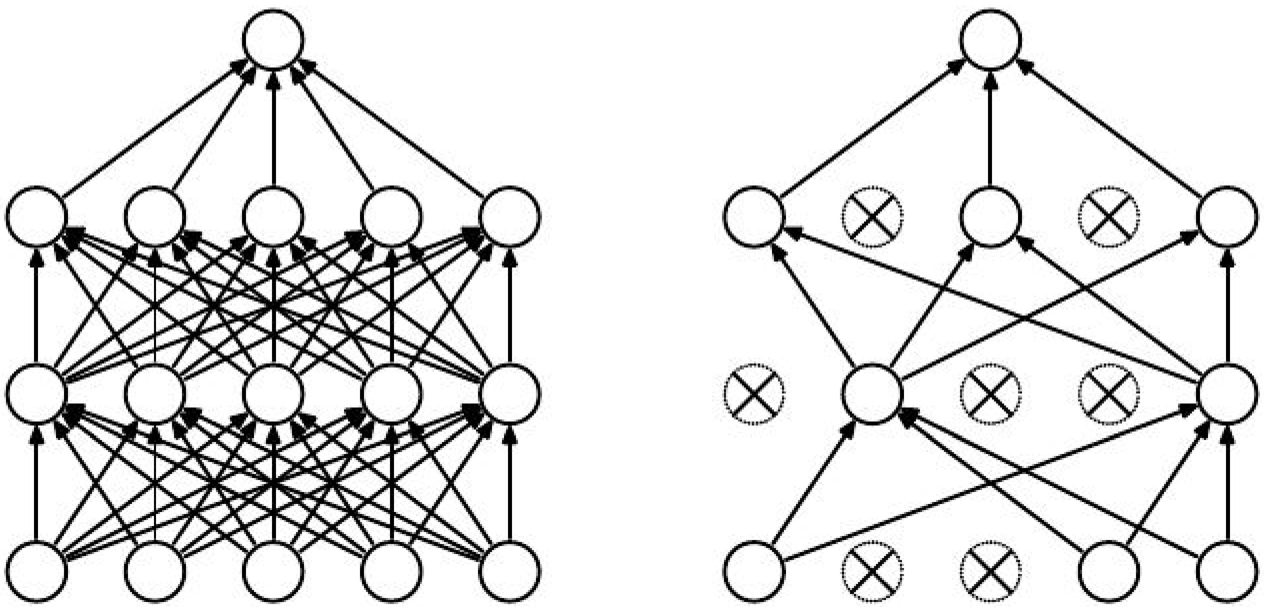
L1 vs L2 regularization

$$\|w\|_1 = \sum_j |w_j| \qquad\qquad \|w\|_2 = \sqrt{\sum_j |w_j|^2}$$



Shi, Jianing V et al. "Perceptual decision making “Through the Eyes” of a large-scale neural model of V1." *Frontiers in psychology* 4 (2013).

Dropout Neural Net Model

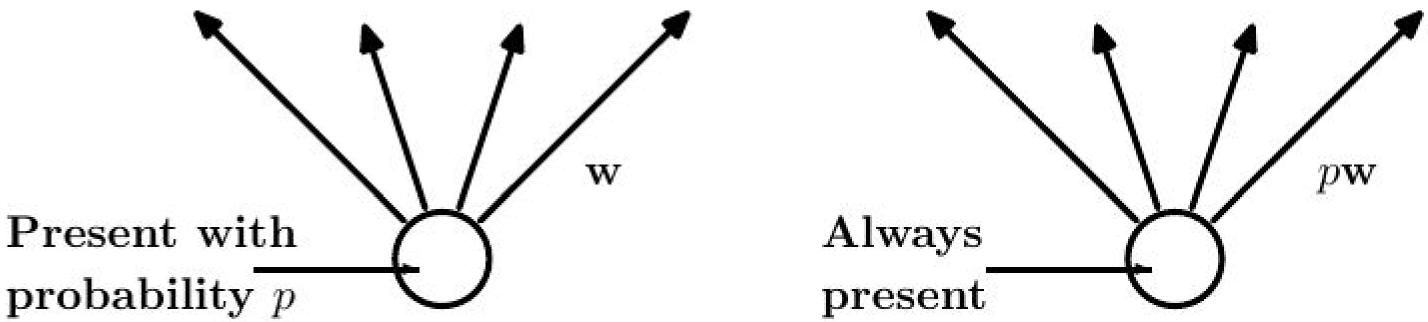


(a) Standard Neural Net　　(b) After applying dropout.

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.

Srivastava, Nitish, et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15 (2014): 1929-1958.

Dropout units



(a) At training time (b) At test time

Left: A unit at training time that is present with probability p and is connected to units in the next layer with weights $\mathbf{w}$. **Right**: At test time, the unit is always present and the weights are multiplied by p. The output at test time is same as the expected output at training time.

Srivastava, Nitish, et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15 (2014): 1929-1958.

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

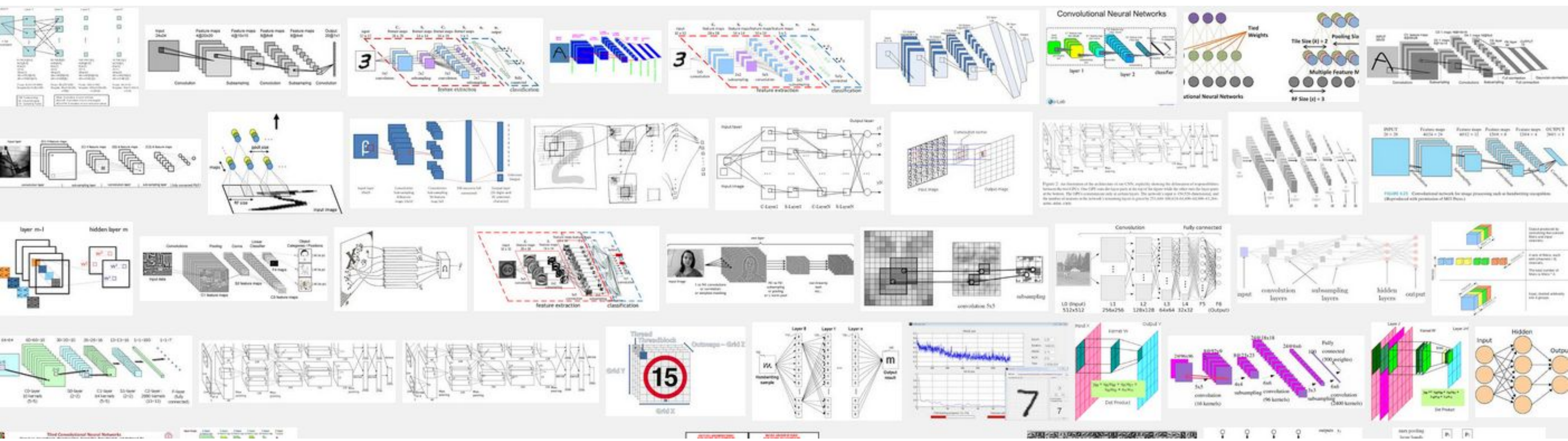
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

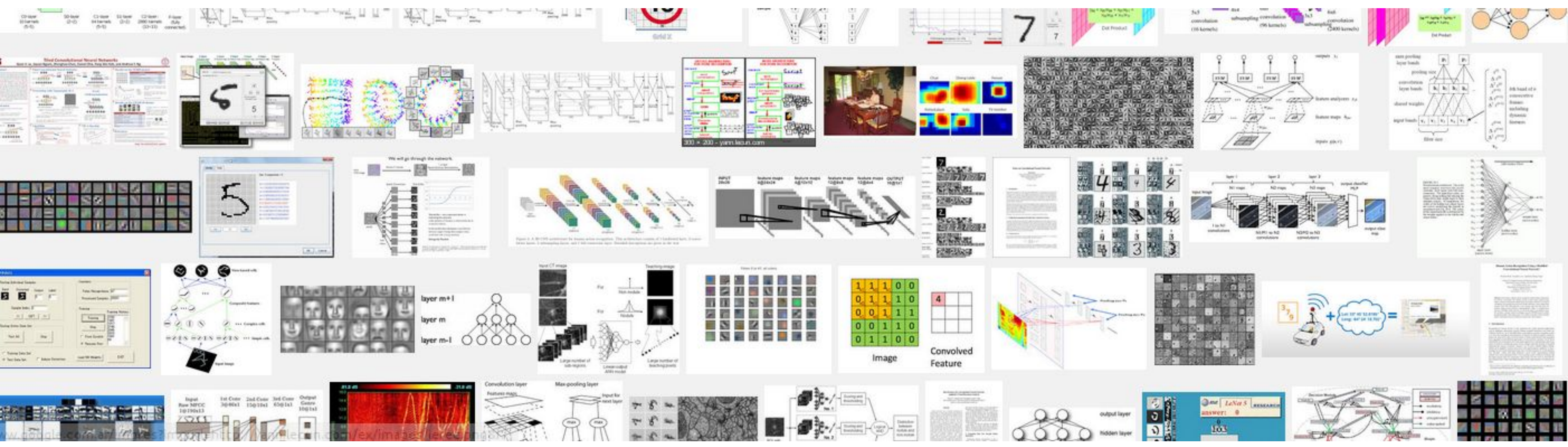
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

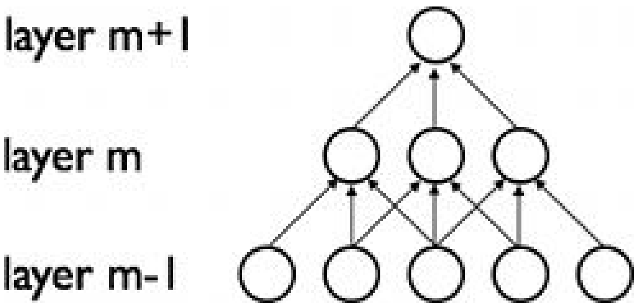


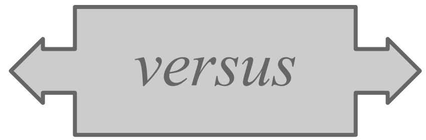
Convolutional Neural Networks (CNN)



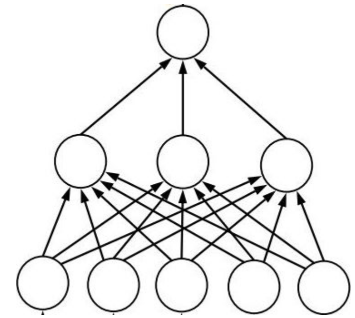
Convolutional Neural Networks

Sparse connectivity

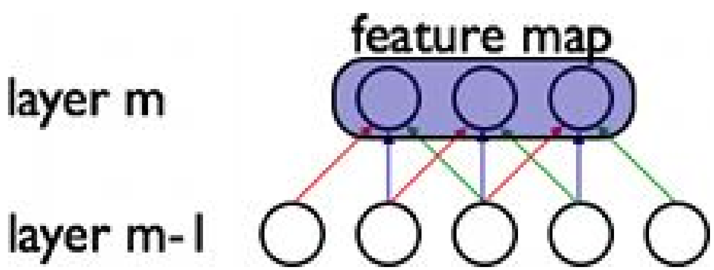




Dense connectivity



Shared weights



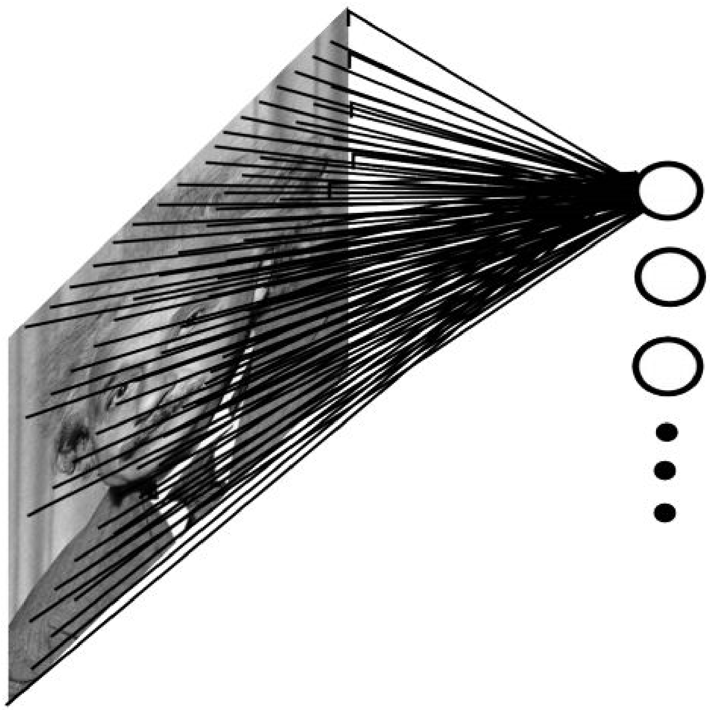
Convolutional Neural Networks

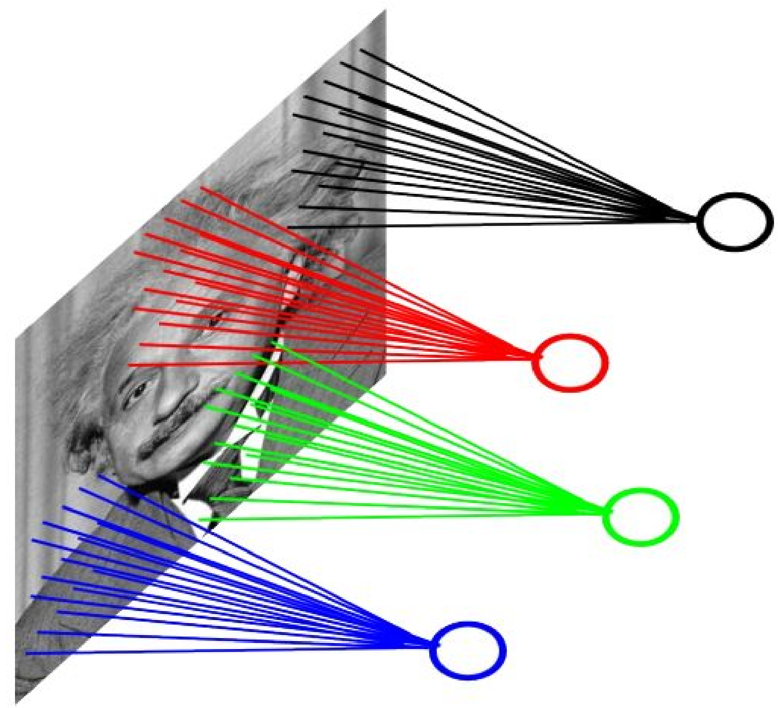
dot product + bias

$$h_k = \tanh(W_k^T x + b_k)$$

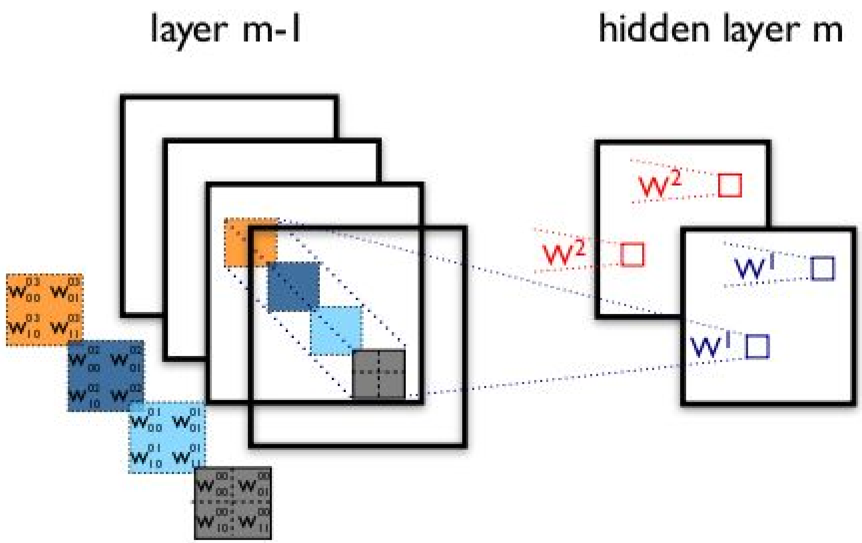
convolution + bias

$$h_{ij}^k = \tanh((W^k * x)_{ij} + b_k)$$





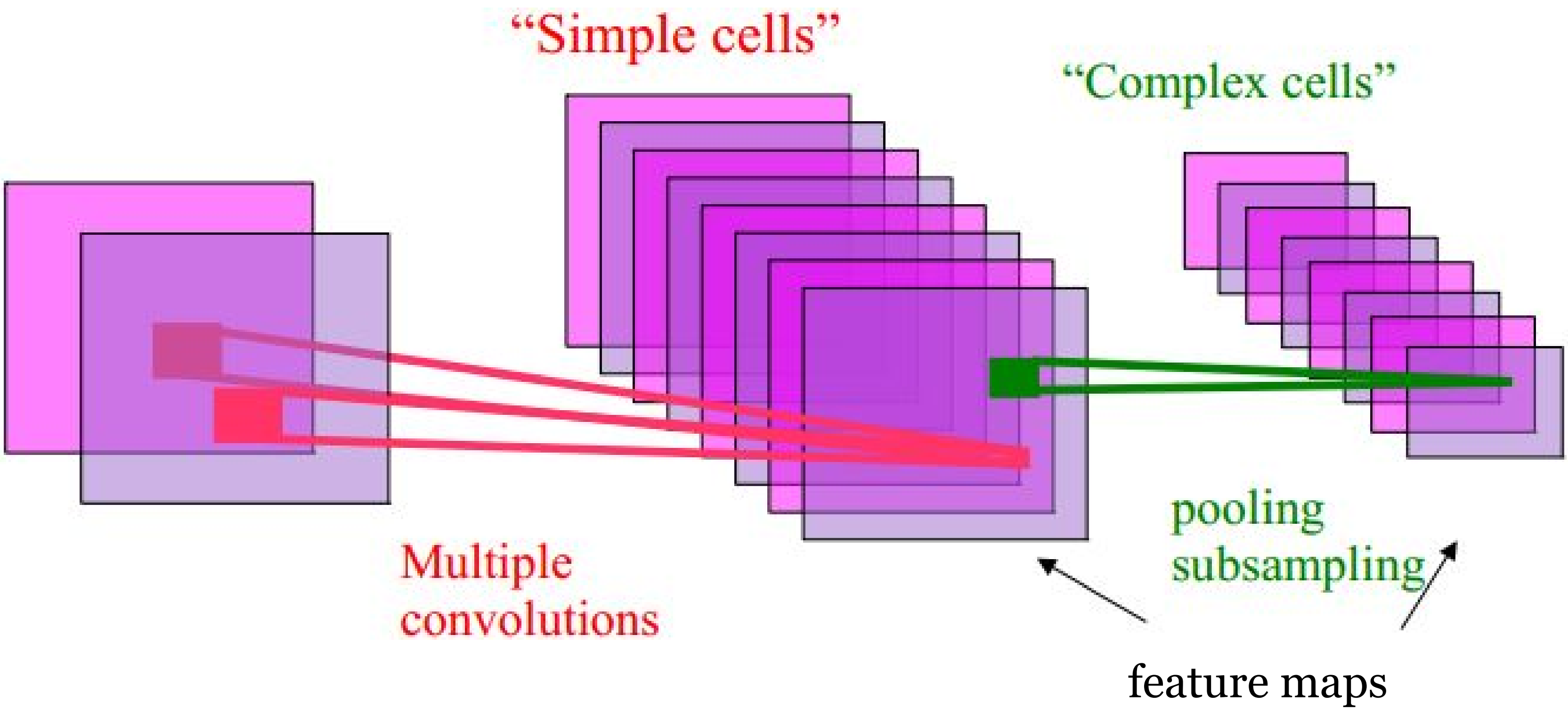
Convolutional Neural Networks



- Detects multiple motifs at each location
- The collection of units looking at the same patch is akin to a feature vector for that patch.
- The result is a 3D array, where each slice is a feature map.

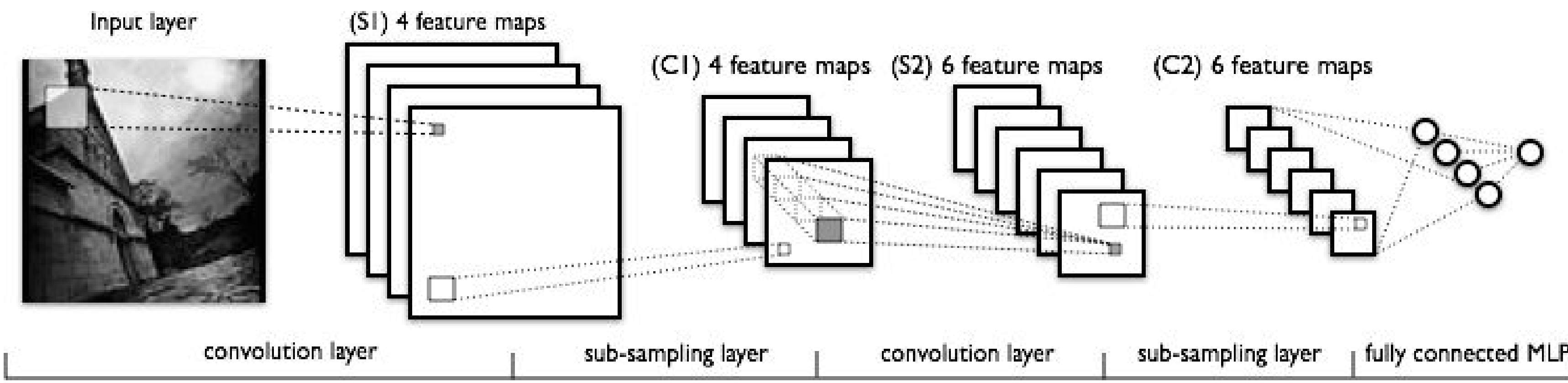
Convolutional Neural Networks

Pooling subsampling



Convolutional Neural Networks

- **Are deployed in many practical applications**
  - Image recognition, speech recognition, Google's and Baidu's photo taggers
- **Have won several competitions**
  - ImageNet, Kaggle Facial Expression, Kaggle Multimodal Learning, German Traffic Signs, Connectomics, Handwriting....
- **Are applicable to array data where nearby values are correlated**
  - Images, sound, time-frequency representations, video, volumetric images, RGB-Depth images,.....
- **One of the few deep models that can be trained purely supervised**

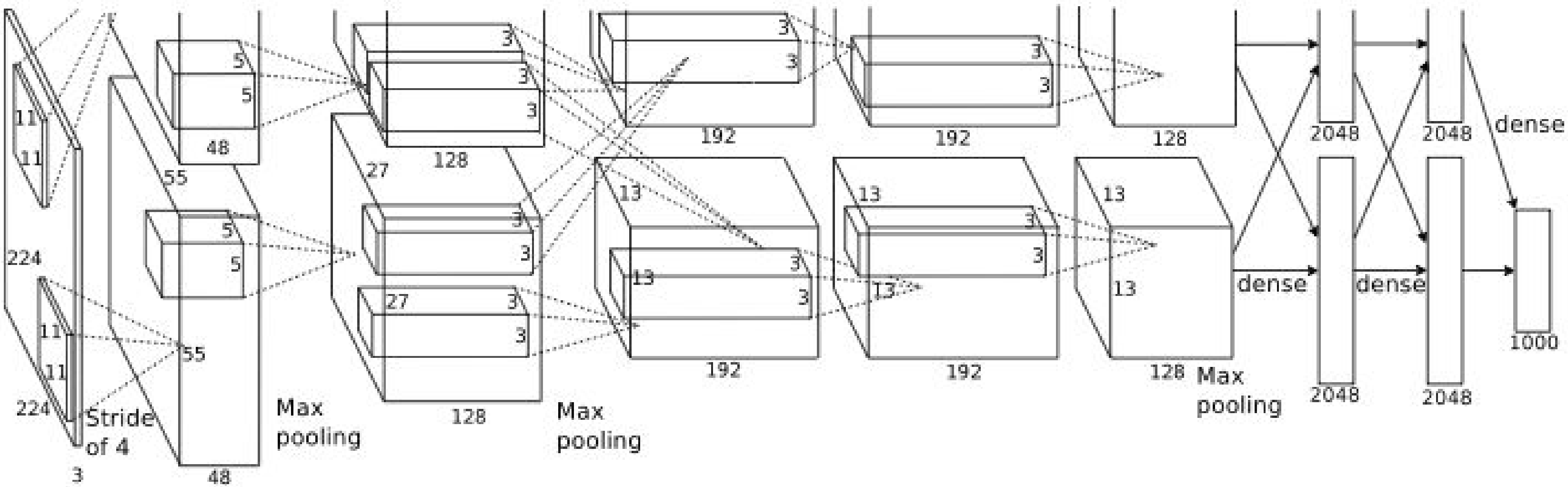


ImageNet Classification with Deep Convolutional Neural Networks

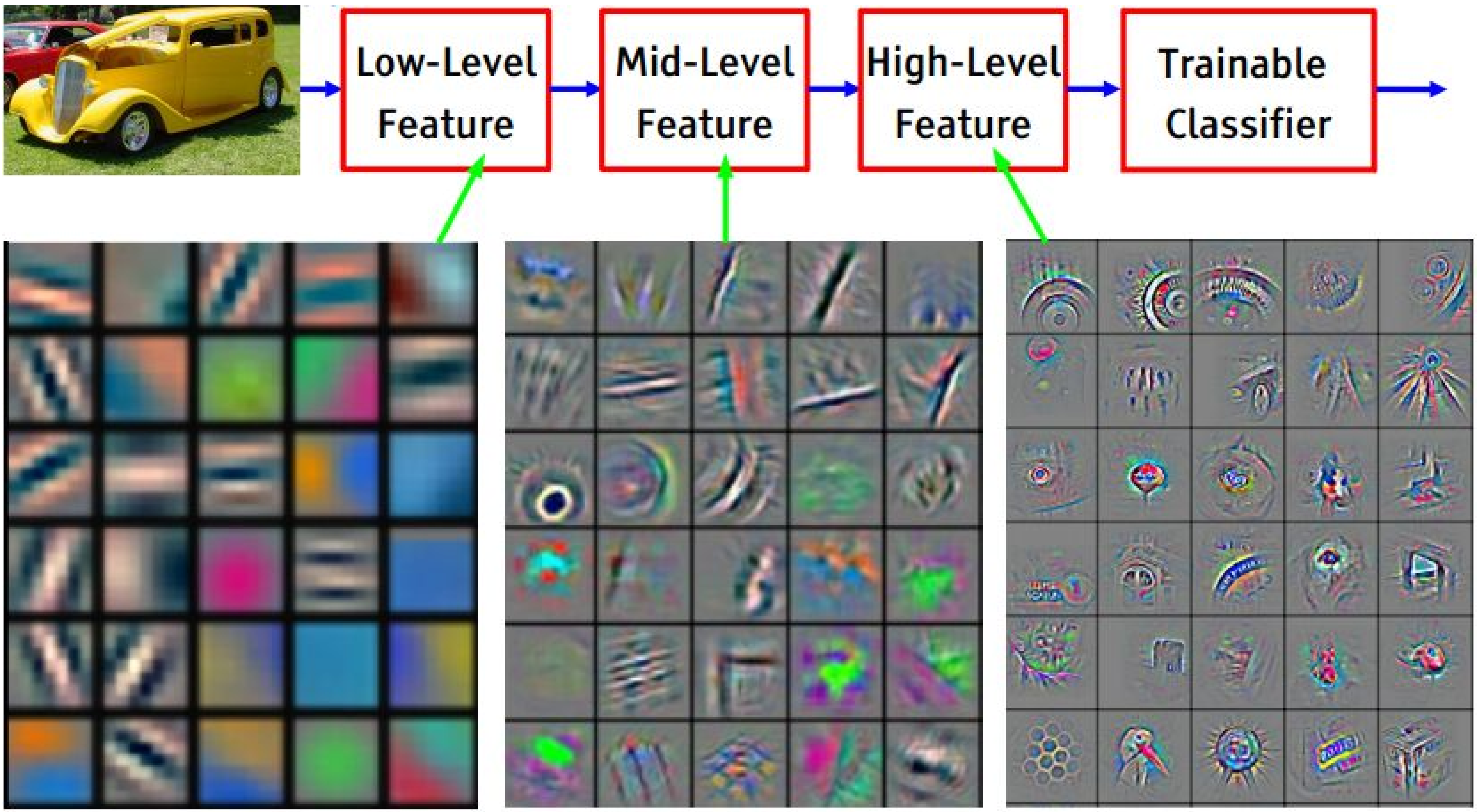
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



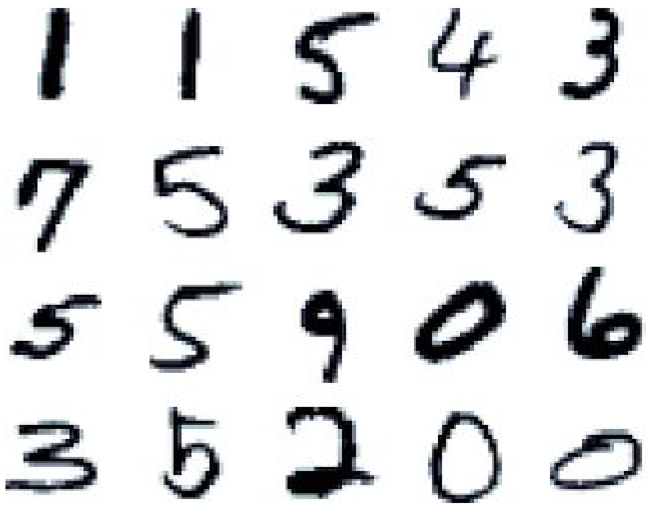
Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 2012: 1097-1105.



Zeiler, Matthew D, and Rob Fergus. "Visualizing and understanding convolutional neural networks." *arXiv preprint arXiv:1311.2901* (2013).

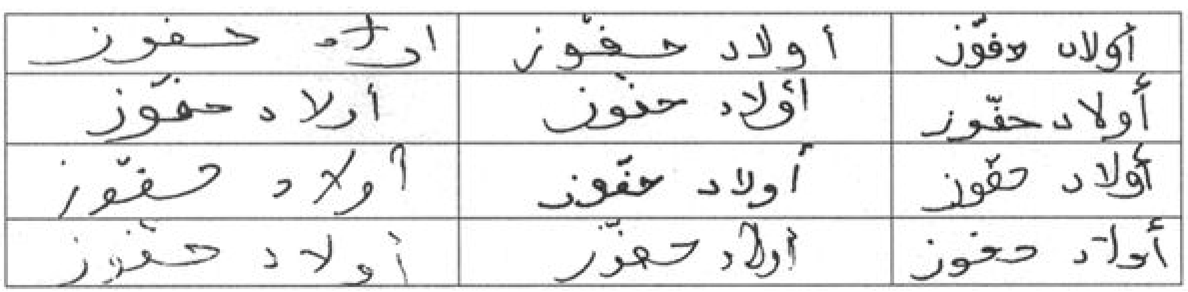
Convolutional Neural Networks

Tasks for Which Deep CNN are the Best



Handwriting recognition MNIST

Arabic Handwriting Recognition



Margner, Volker, and Haikal El Abed. "Arabic handwriting recognition competition." *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* 23 Sep. 2007: 1274-1278.

Tasks for Which Deep CNN are the Best

StreetView House Numbers [2011]

94.3 % accuracy

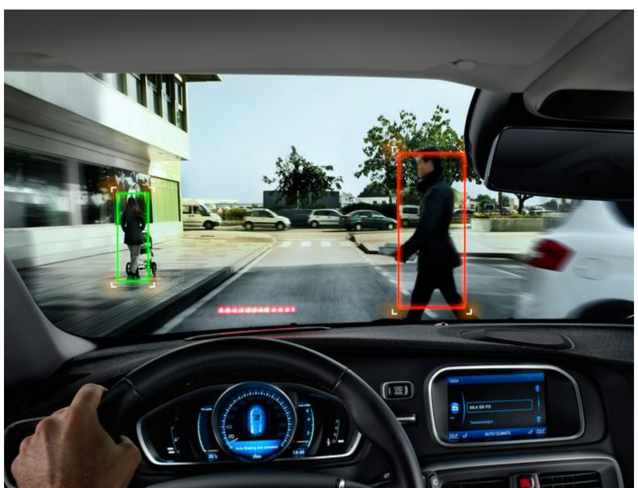
Netzer, Yuval et al. "Reading digits in natural images with unsupervised feature learning." *NIPS workshop on deep learning and unsupervised feature learning* 2011: 4.

Tasks for Which Deep CNN are the Best

Traffic Sign Contest, Silicon Valley, 2011 (IDSIA)

0.56% ERROR

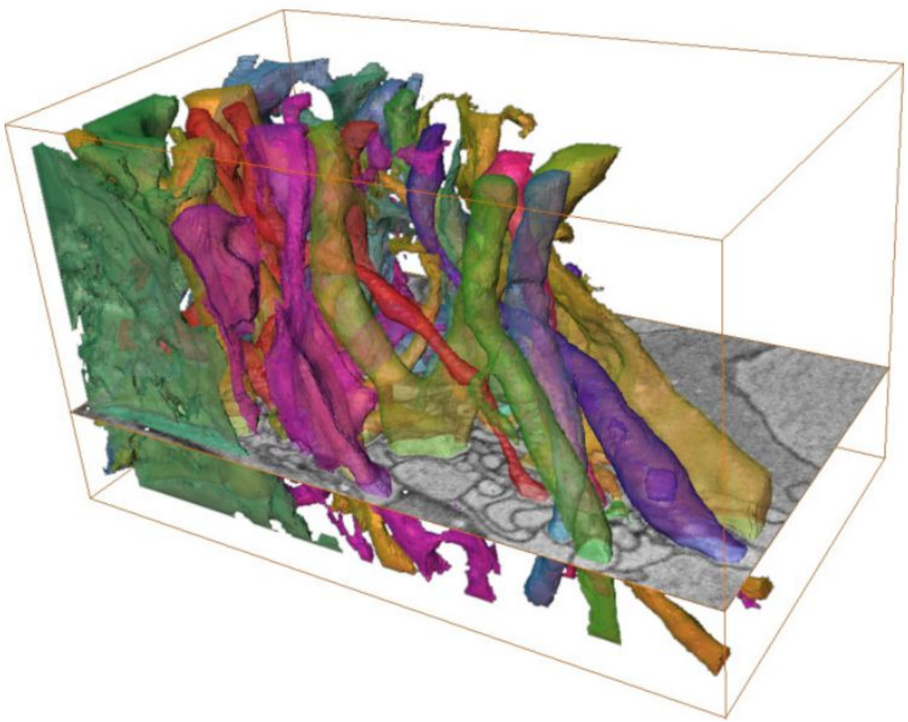
- first place
- twice better than humans
- three times better than the closest artificial competitor
- six times better than the best non-neural method



Pedestrian Detection [2013]: INRIA datasets and others (NYU)

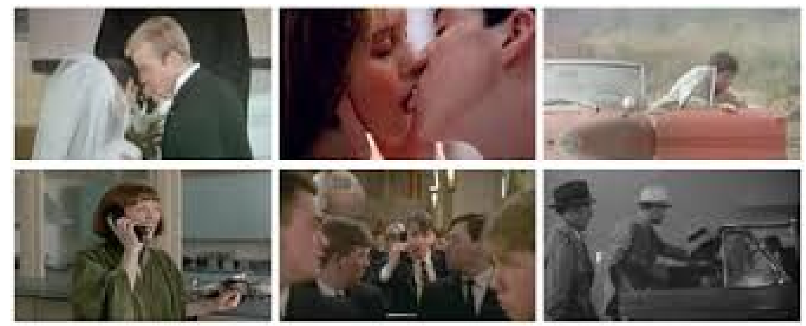
Tasks for Which Deep CNN are the Best

Volumetric brain image segmentation [2009] Connectomics (IDSIA, MIT)



Turaga, Srinivas C et al. "Convolutional networks can learn to generate affinity graphs for image segmentation." *Neural Computation* 22.2 (2010): 511-538.

Human Action Recognition [2011] Hollywood II dataset (Stanford)



Le, Quoc V et al. "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis." *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* 20 Jun. 2011: 3361-3368.

Tasks for Which Deep CNN are the Best

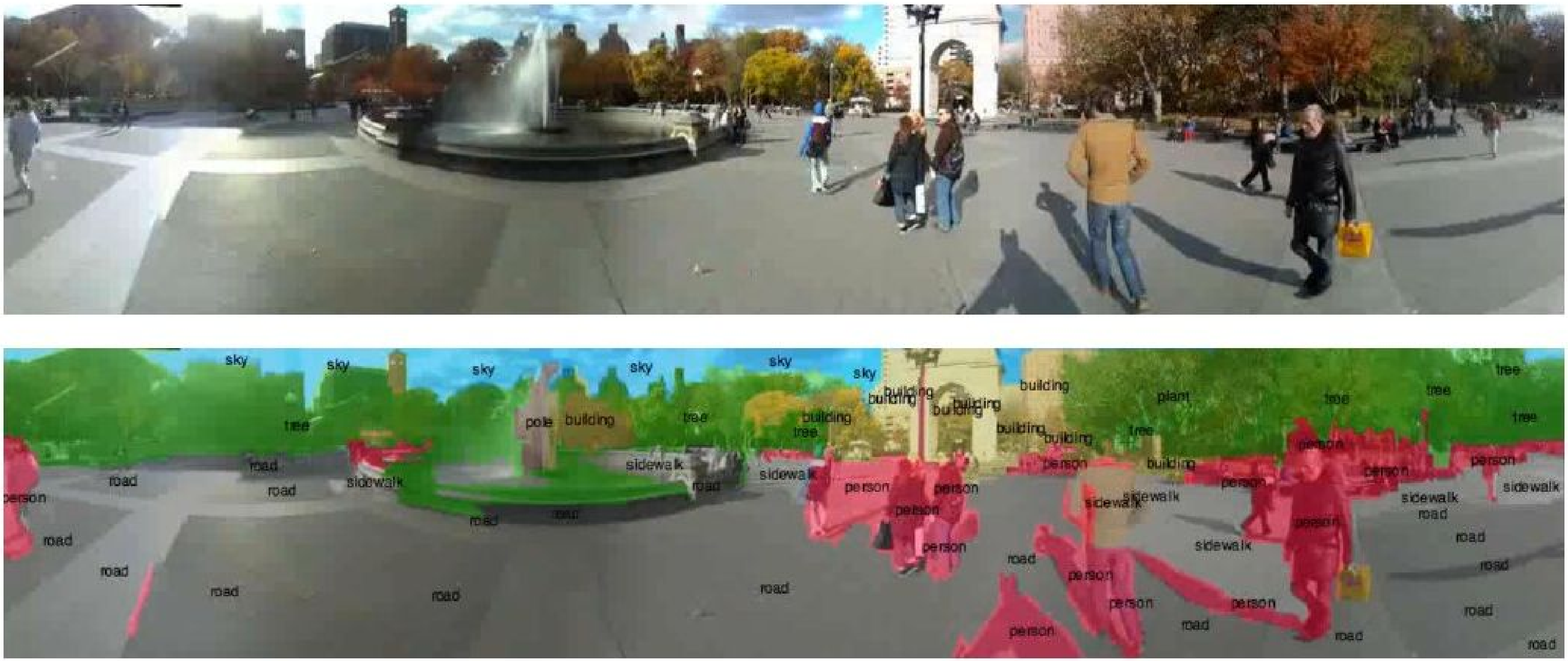
Object Recognition [2012] ImageNet competition

Error rate: 15% (whenever correct class isn't in top 5)
Previous state of the art: 25% error

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 2012: 1097-1105.

Tasks for Which Deep CNN are the Best

Scene Parsing [2012]



Farabet, Clément et al. "Scene parsing with multiscale feature learning, purity trees, and optimal covers." *arXiv preprint arXiv:1202.2160* (2012).

Google AI algorithm masters ancient game of Go

Deep-learning software defeats human professional for first time.

27 January 2016

A computer has beaten a human professional for the first time at Go — an ancient board game that has long been viewed as one of the greatest challenges for artificial intelligence (AI)

"We pass in the board position as a 19 × 19 image and use convolutional layers to construct a representation of the position."

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489.

Andrej Karpathy blog

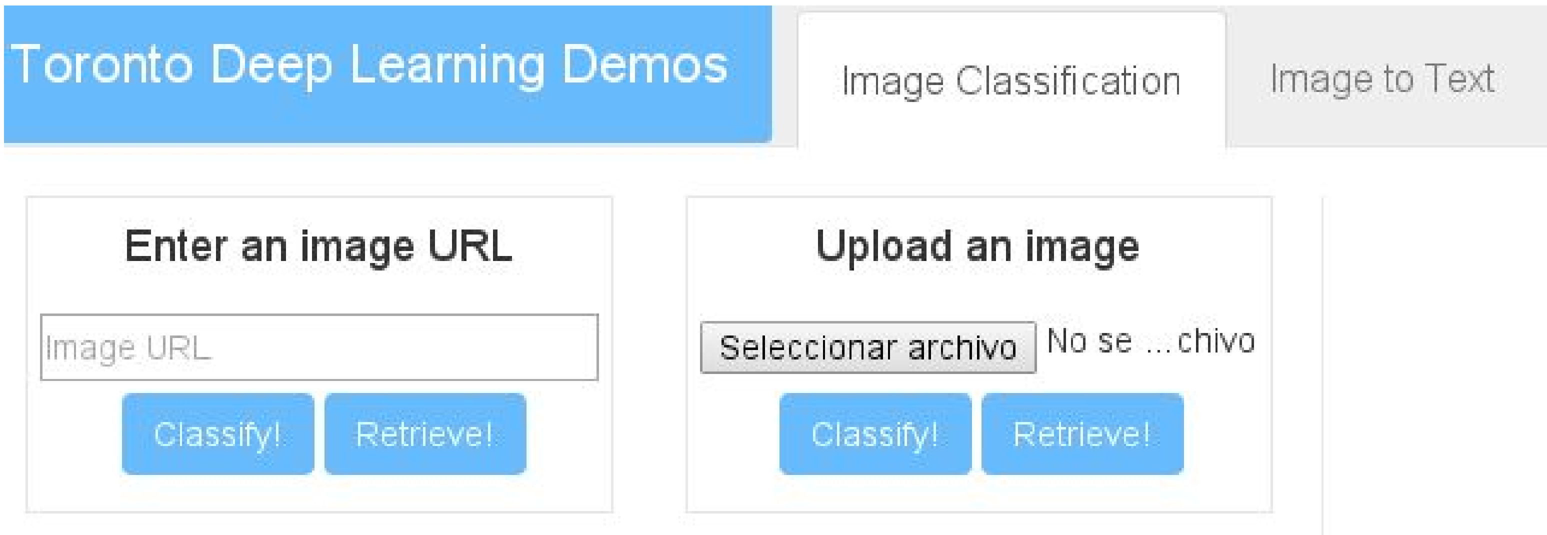
About Hacker's guide to Neural Networks

What a Deep Neural Network thinks about your #selfie

Oct 25, 2015

http://karpathy.github.io/2015/10/25/selfie/

http://deeplearning.cs.toronto.edu/



DNN Frameworks

Python:

- **Theano** Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently
- **Pylearn2** Machine learning library. Most of its functionality is built on top of Theano.
- **Keras** Keras is a minimalist, highly modular neural networks library, written in Python and capable of running on top of either TensorFlow or Theano.
- **Veles** Distributed platform for rapid Deep learning application development.

Python & C++:

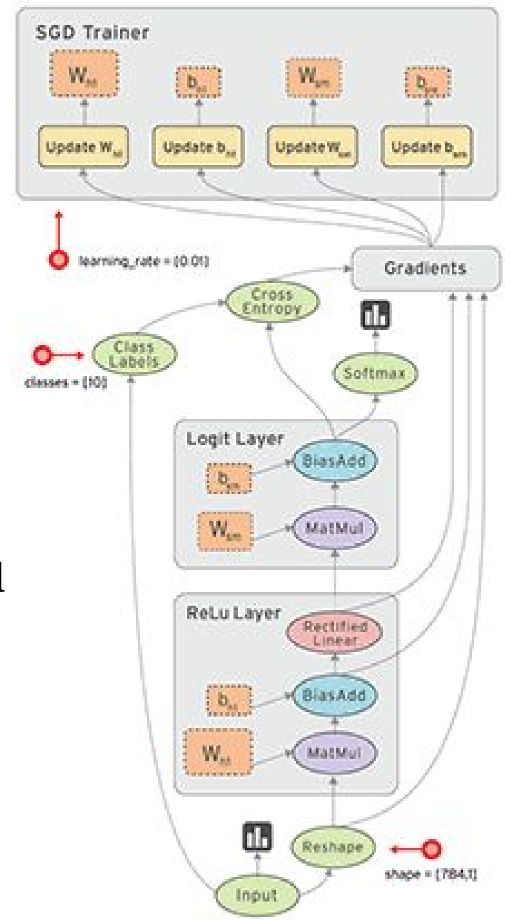
- **TensorFlow** Library for numerical computation using data flow graphs.
- **Caffe** Deep learning framework made with expression, speed, and modularity in mind.

Lua:

- **Torch7** Scientific computing framework with wide support for machine learning algorithms

C++:

- **Marvin** A minimalist GPU-only N-dimensional ConvNet framework



Pretrained models

Torch7 Model Zoo:

- Overfeat
- DeepCompare

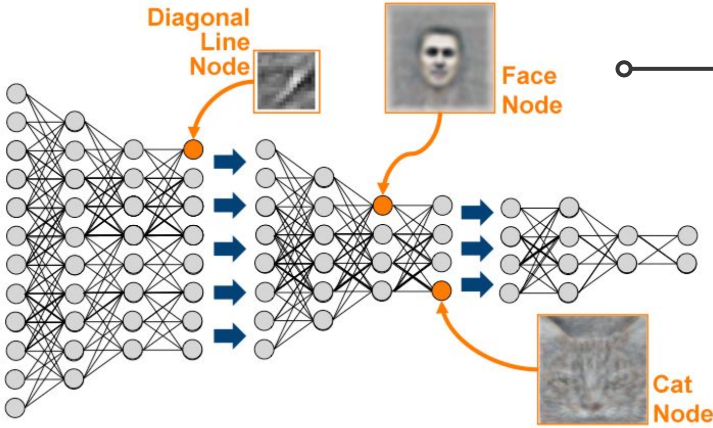
Caffe Model Zoo

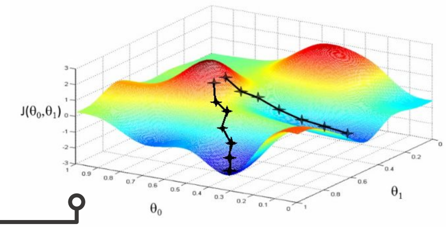
- Models for Age and Gender Classification
- Image Segmentation
- Translating Videos to Natural Language
- ...

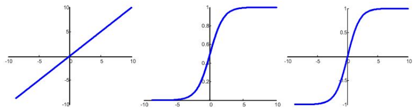
Marvin:

- AlexNet
- GoogLeNet
- VGG16

TensorFlow: TODO.







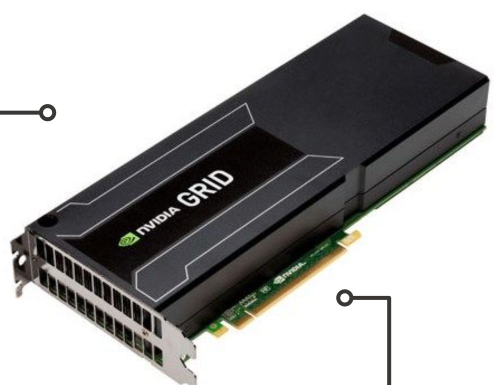
RIO2016

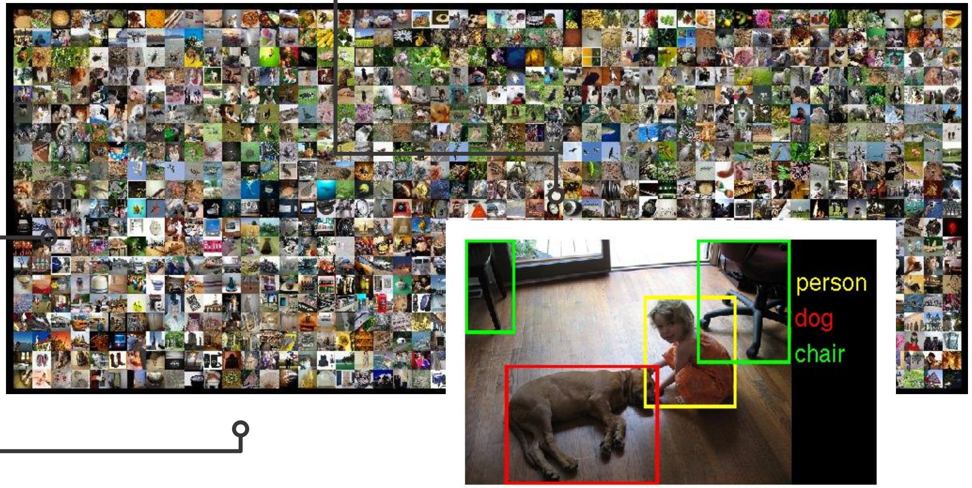
23º Escuela de Verano
de Ciencias Informáticas

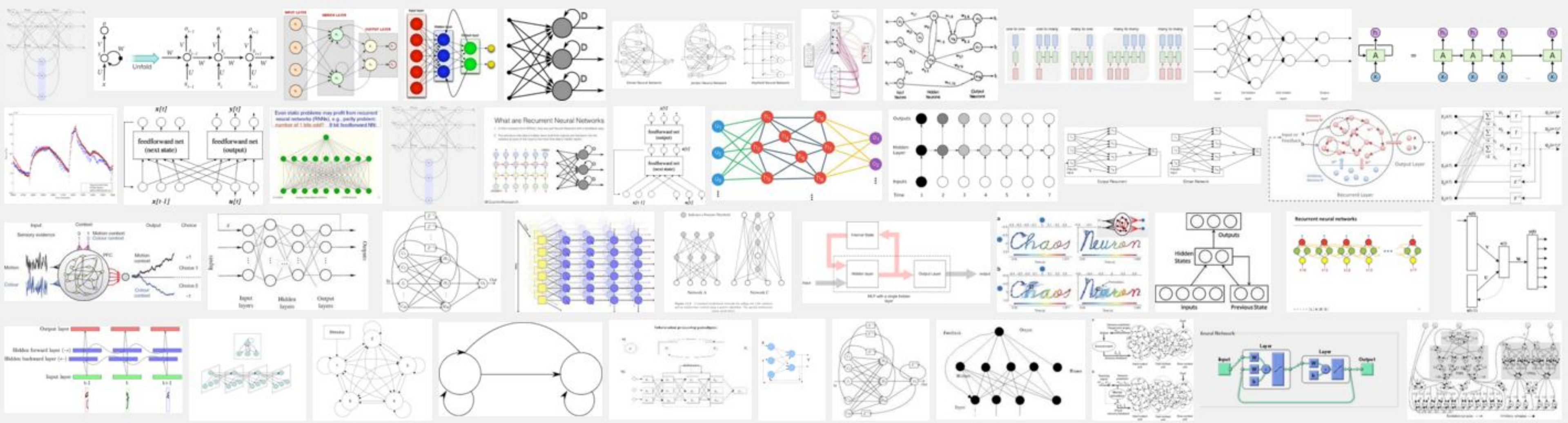
Métodos Actuales de Machine Learning

Neural Networks

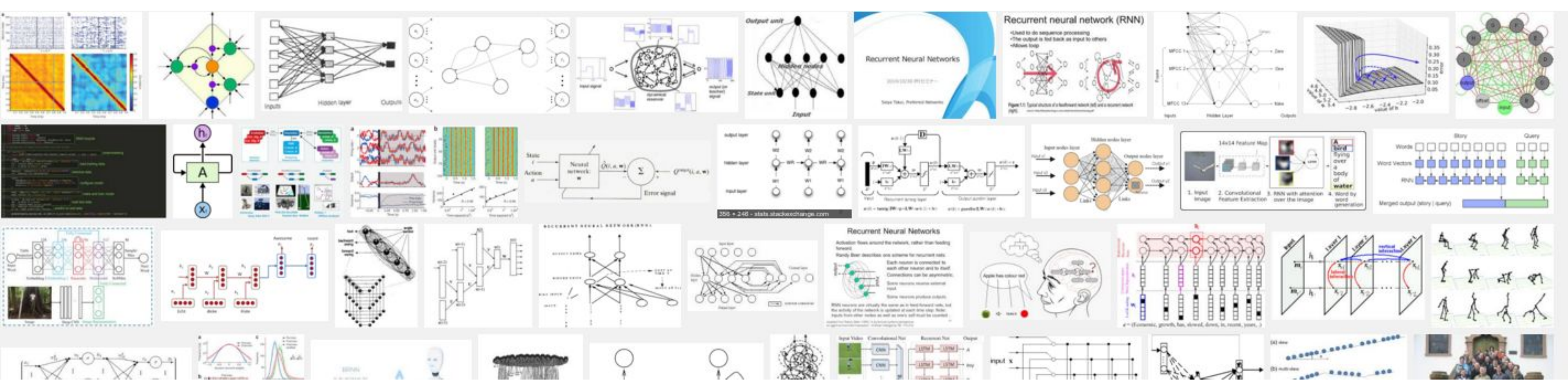
http://www.cifasis-conicet.gov.ar/granitto/RIO2016/





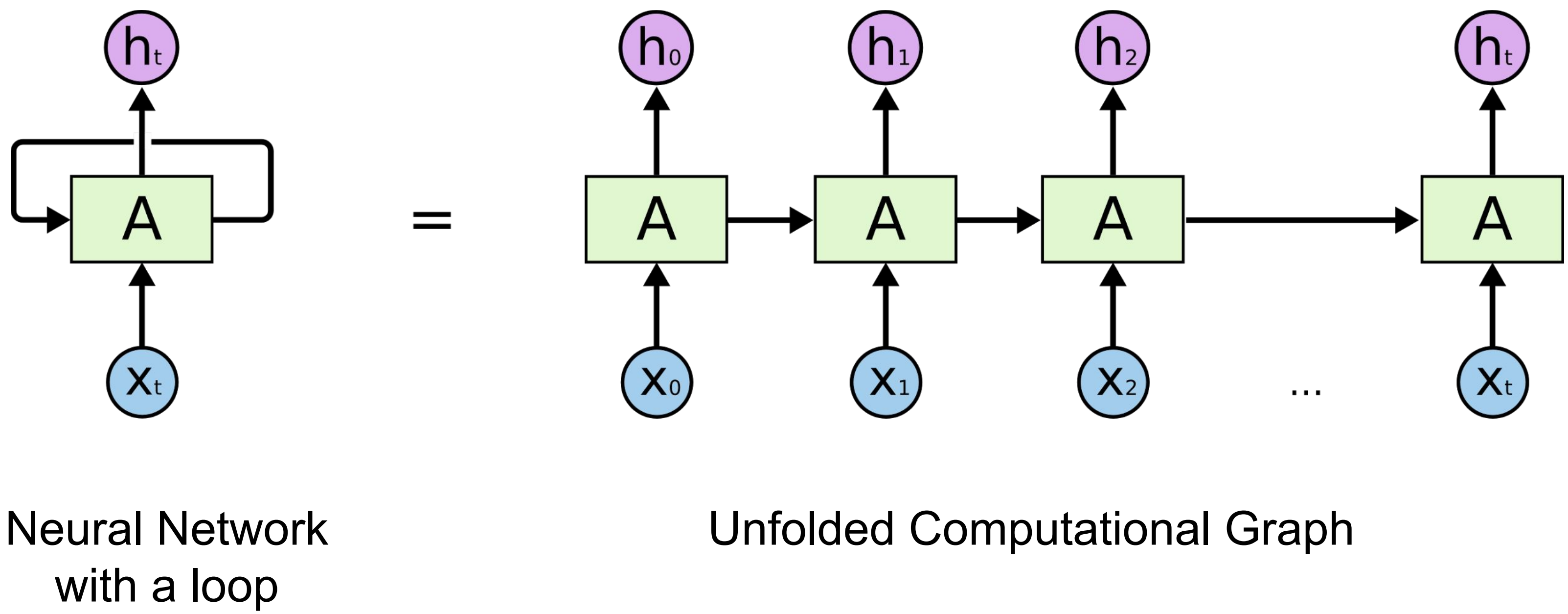


Recurrent Neural Networks

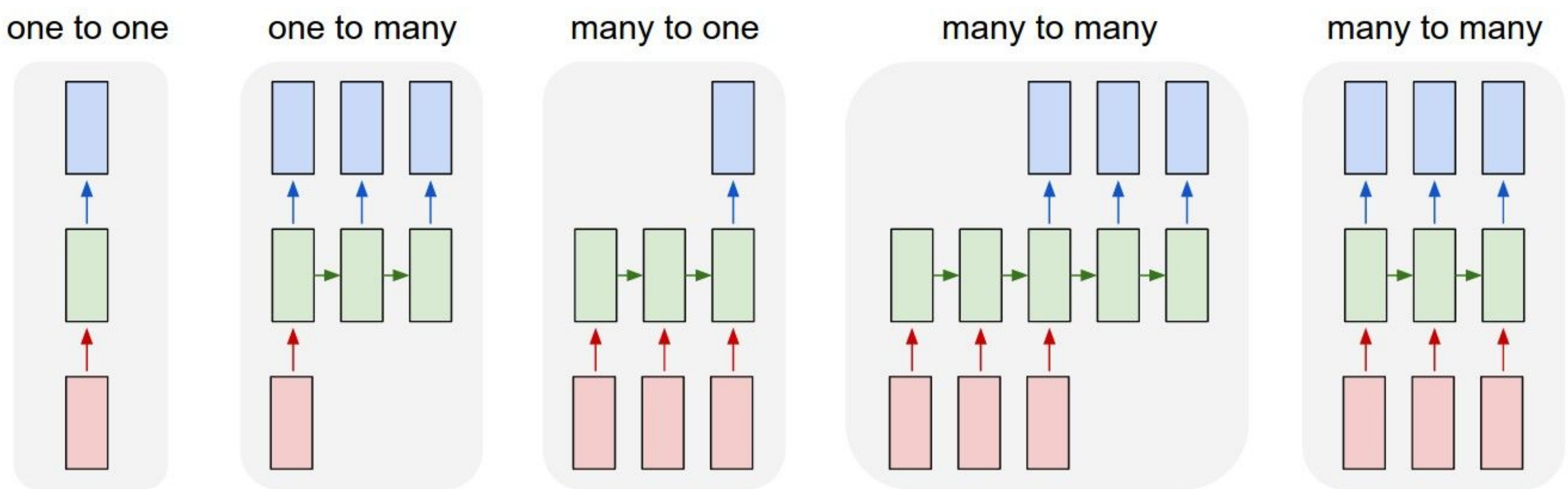


Recurrent Neural Networks

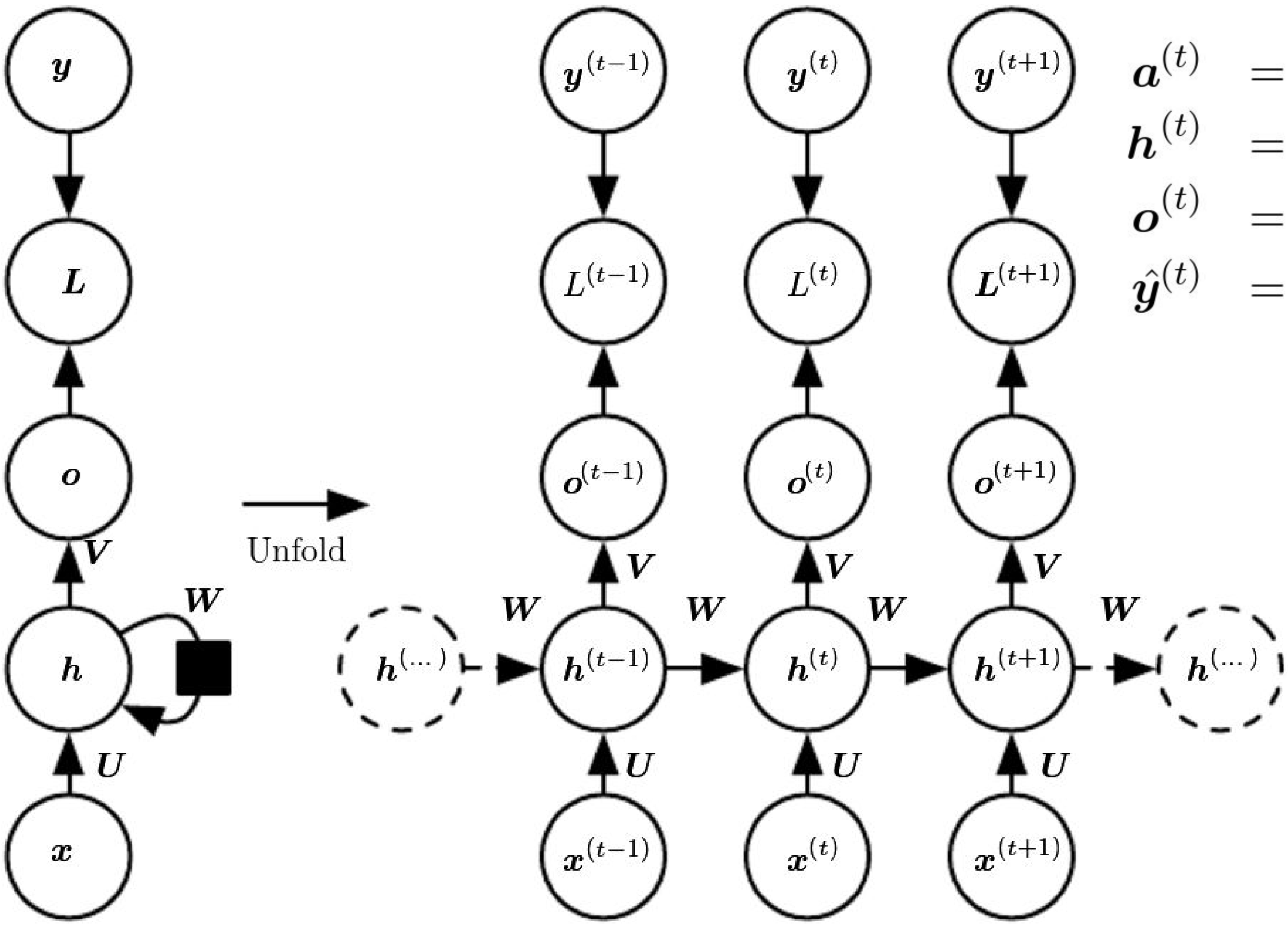
from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/



Recurrent Neural Networks



Recurrent Neural Networks

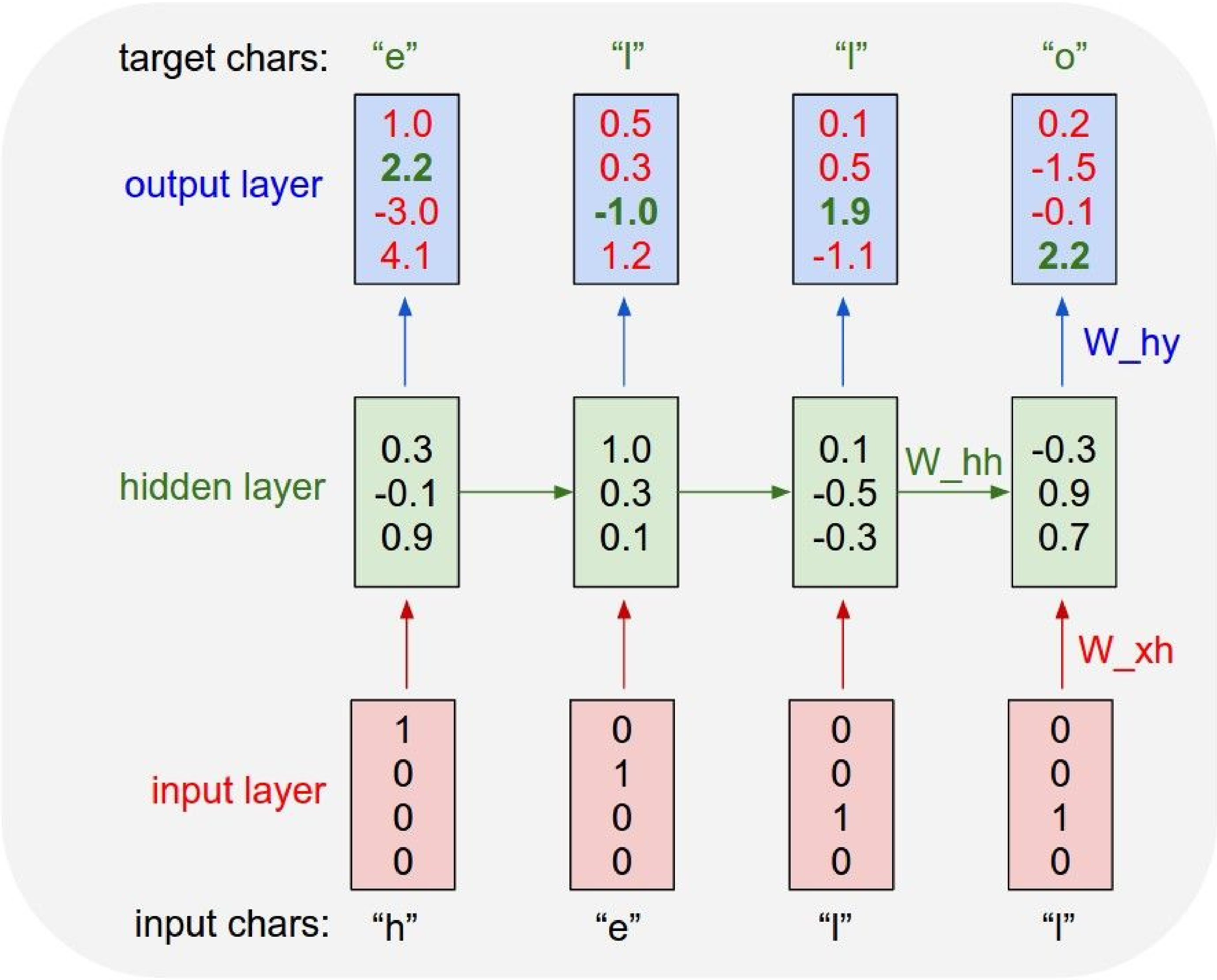


$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \mathrm{softmax}(\boldsymbol{o}^{(t)})
\end{aligned}
$$

Deep Learning (Book in preparation) Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016

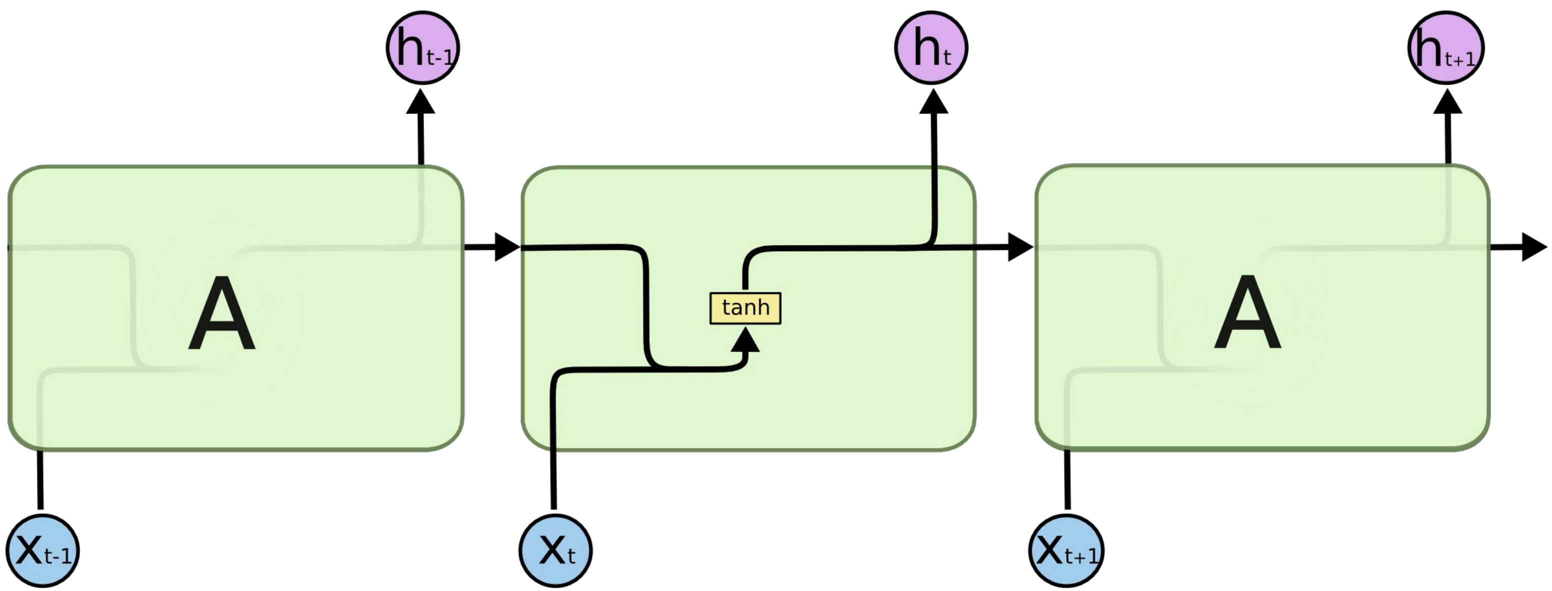
Recurrent Neural Networks

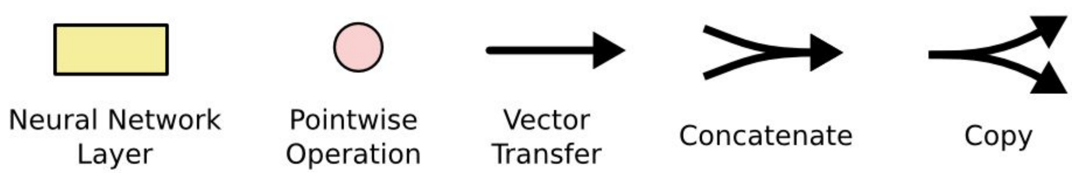
from Andrej Karpathy blog http://karpathy.github.io/2015/05/21/rnn-effectiveness/



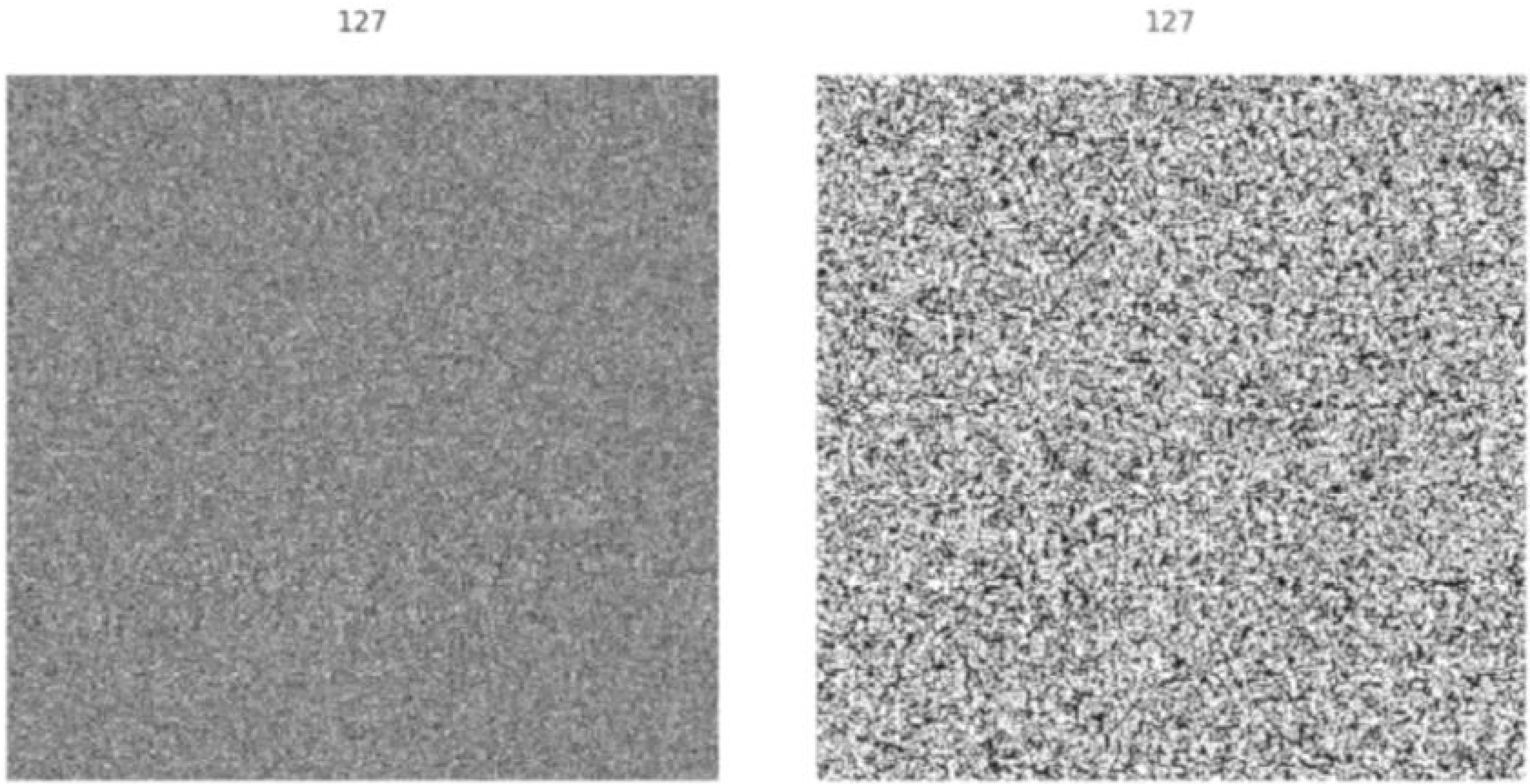
Recurrent Neural Networks

from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/



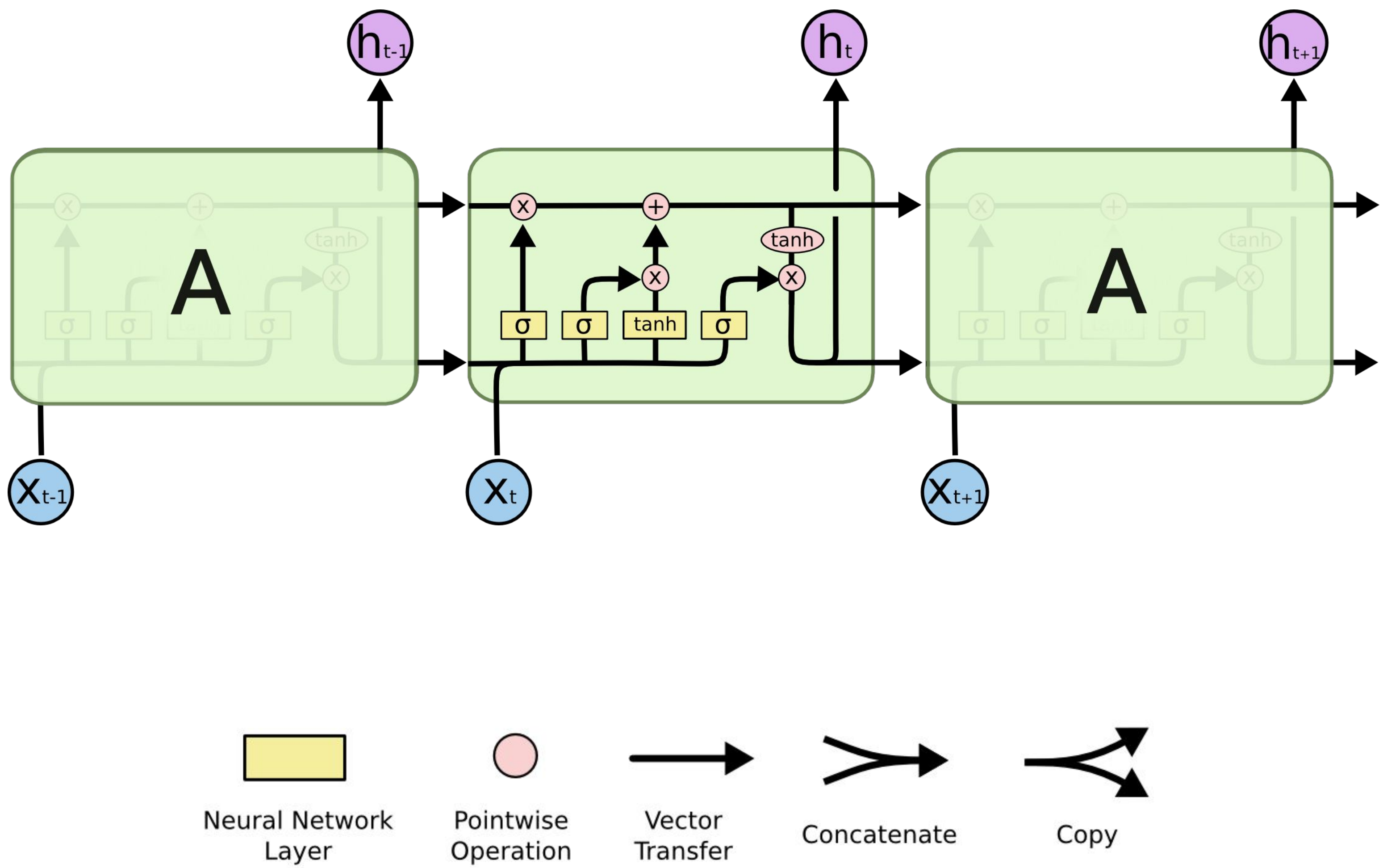


RNN: Vanishing Gradient Problem



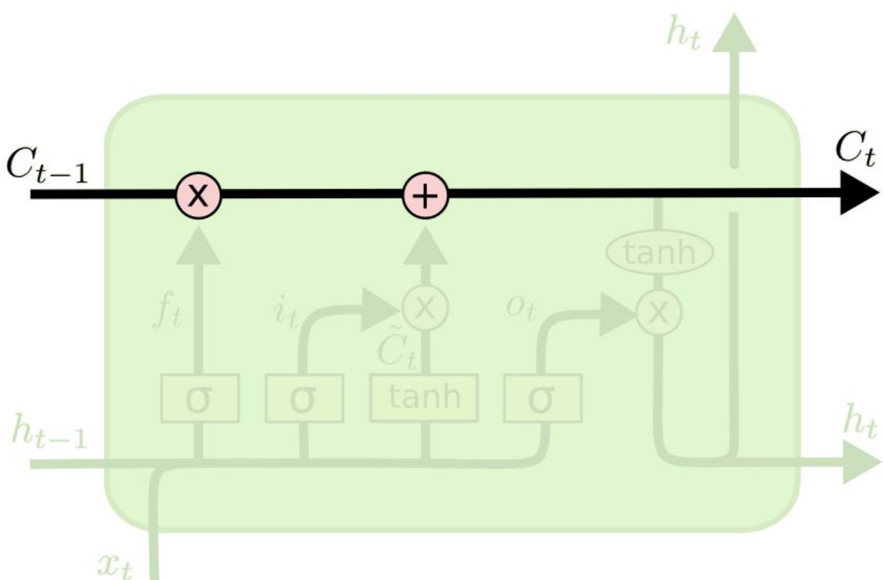
LSTM: Long Short-Term Memory

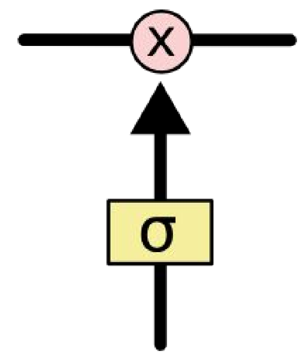
from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/

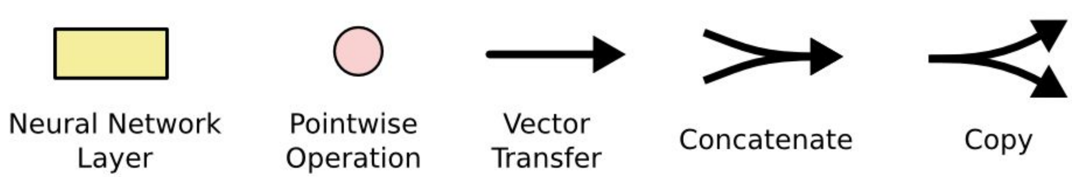


LSTM: Long Short-Term Memory

from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/

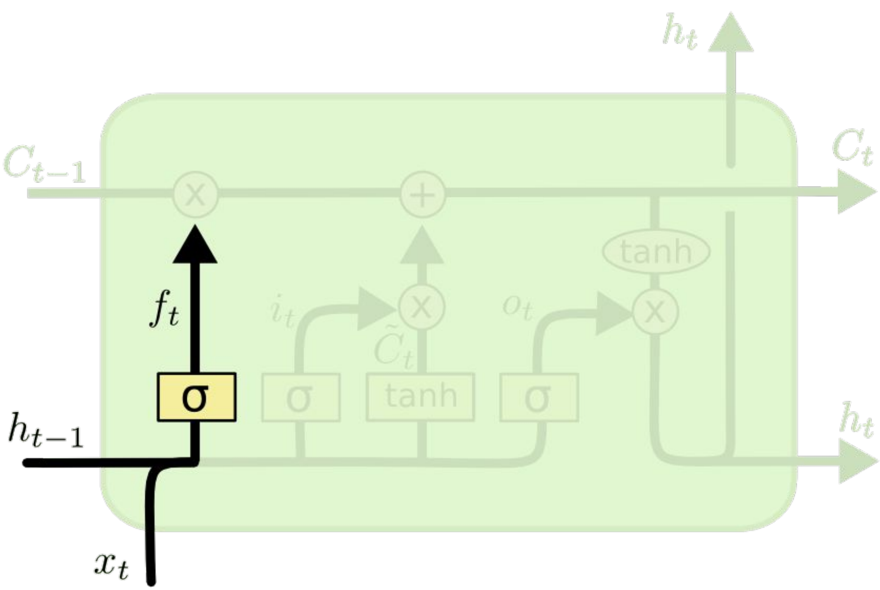




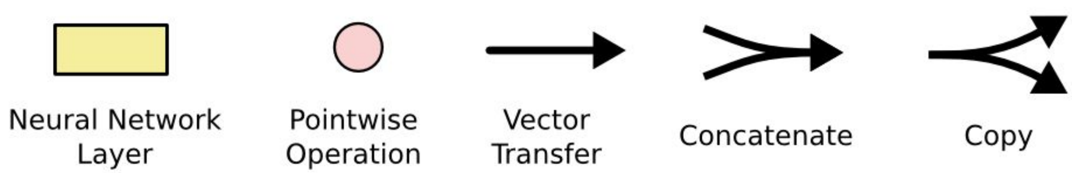


LSTM: Long Short-Term Memory

from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/

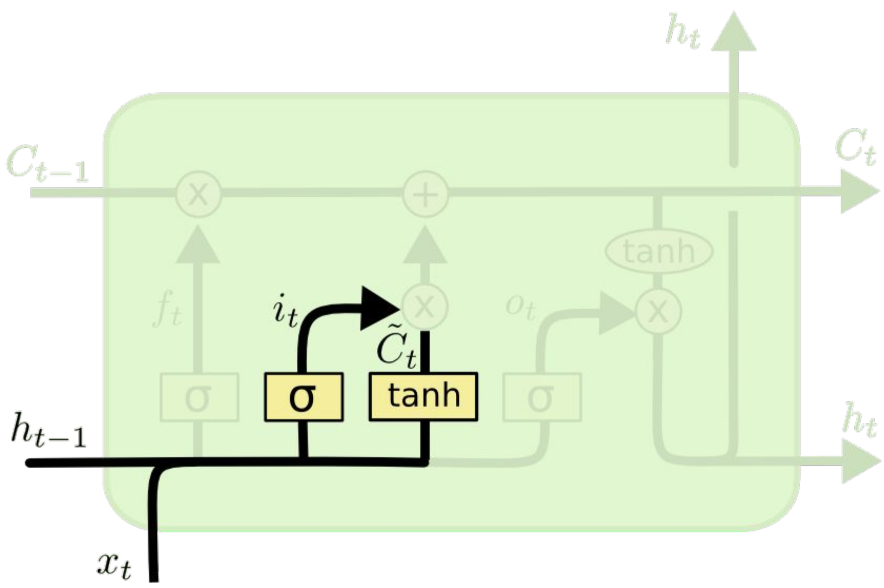


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



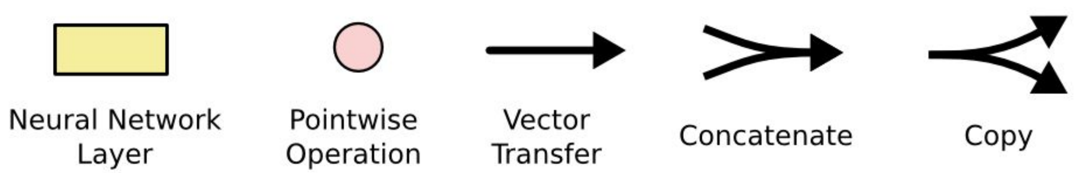
LSTM: Long Short-Term Memory

from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/



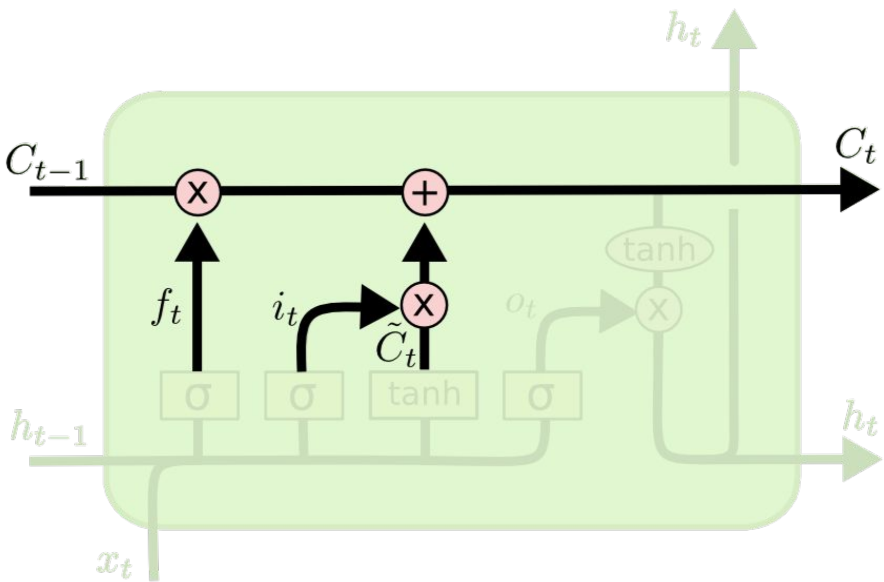
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

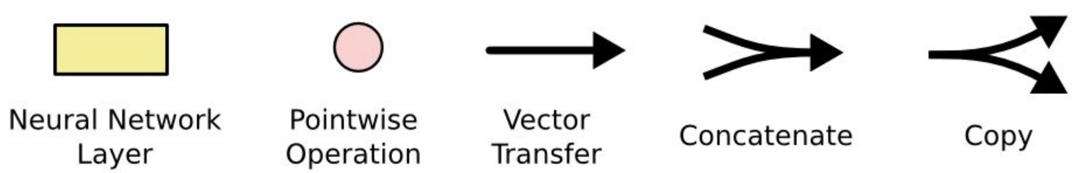


LSTM: Long Short-Term Memory

from colah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/

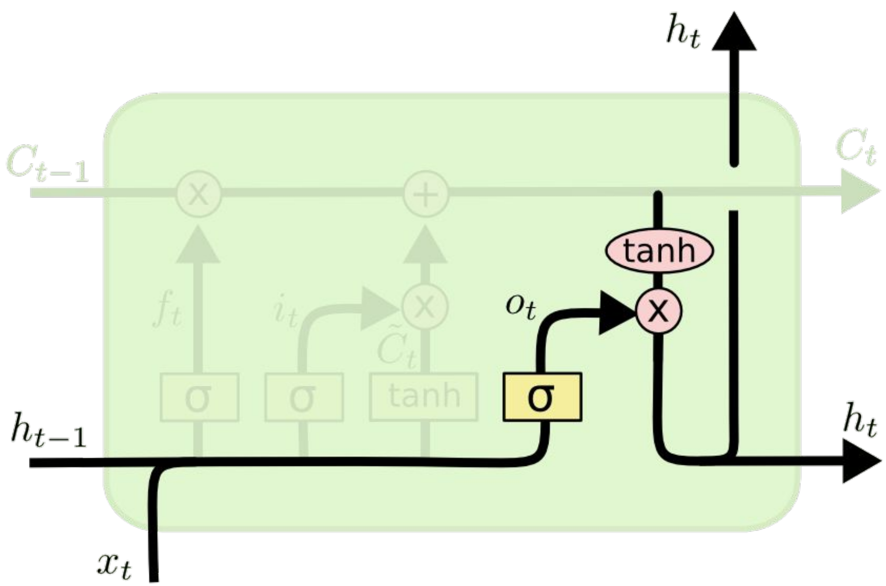


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



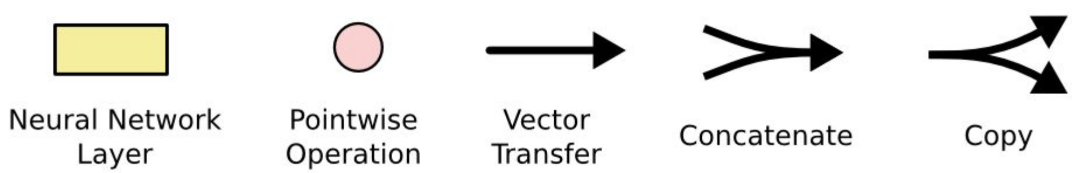
LSTM: Long Short-Term Memory

from [colah's blog](http://colah.github.io/posts/2015-08-Understanding-LSTMs/) http://colah.github.io/posts/2015-08-Understanding-LSTMs/



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$



LSTM: Sampled Wikipedia articles

from [Andrej Karpathy blog] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]]
(PJS)[http://www.humah.yahoo.com/guardian.
cfm/7754800786d17551963s89.htm Official economics Adjoint for the Nazism, Montgomery
was swear to advance to the resources for those Socialism's rule,
was starting to signing a major tripad of aid exile.]]

LSTM: Sampled Linux source code

from [Andrej Karpathy blog](http://karpathy.github.io/2015/05/21/rnn-effectiveness/) http://karpathy.github.io/2015/05/21/rnn-effectiveness/

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

Deep Visual-Semantic Alignments for Generating Image Descriptions

from http://cs.stanford.edu/people/karpathy/deepimagesent/

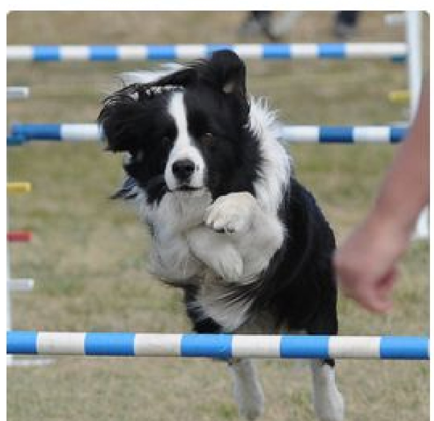
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."



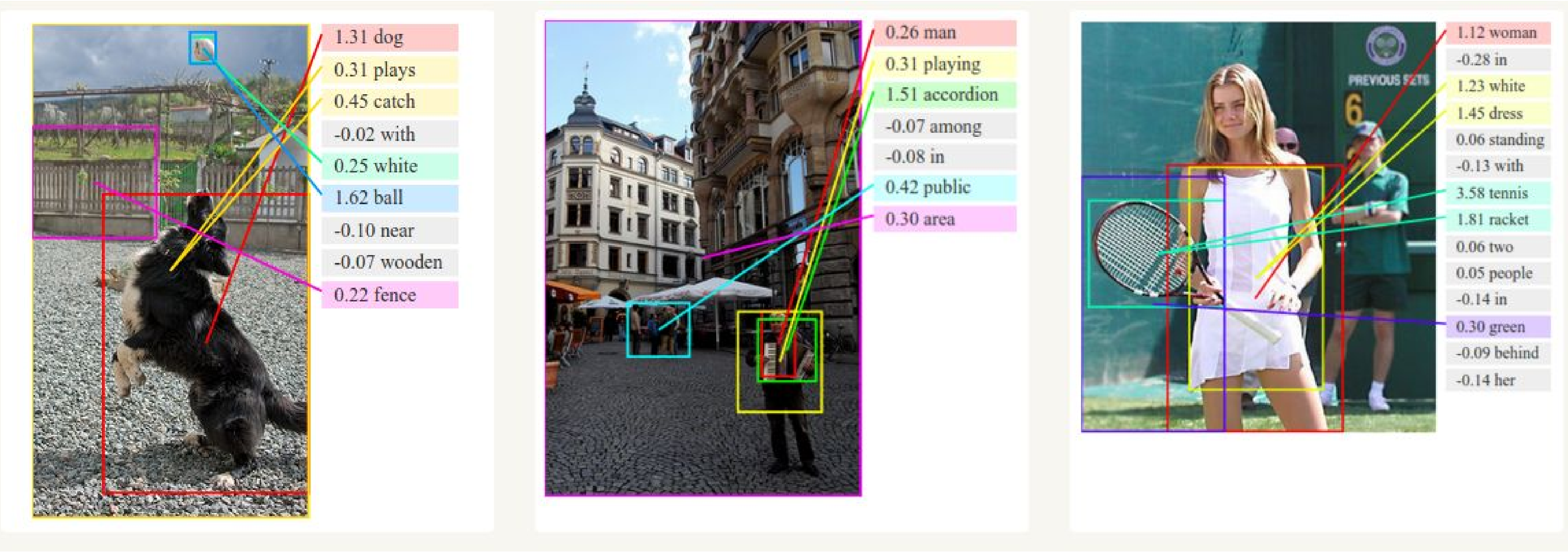
"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

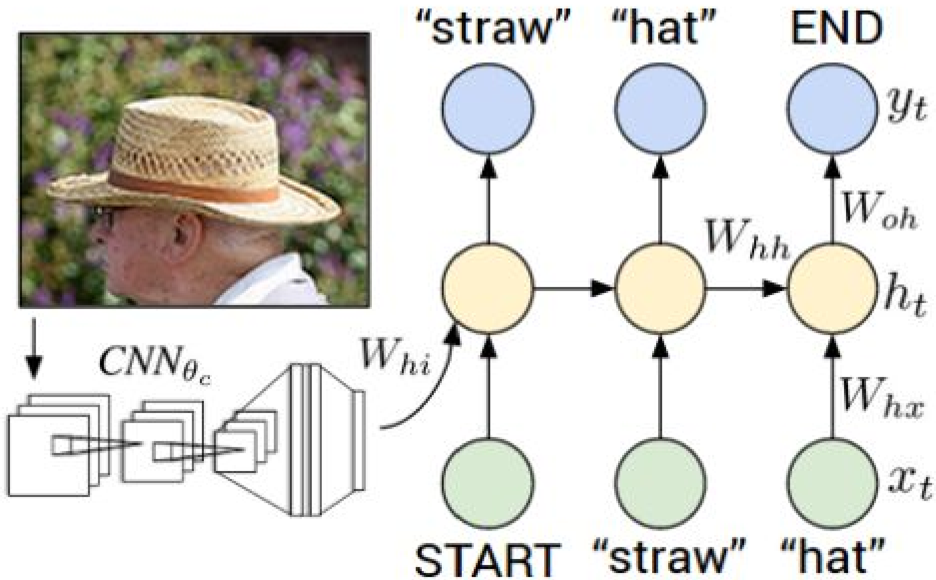
Deep Visual-Semantic Alignments for Generating Image Descriptions

from http://cs.stanford.edu/people/karpathy/deepimagesent/



Deep Visual-Semantic Alignments for Generating Image Descriptions

from http://cs.stanford.edu/people/karpathy/deepimagesent/



Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.