**EM method:**

Algorithm for mixture models (like MOG), useful for funding the parameters (mean,variance) of the Geussians that describes the data.

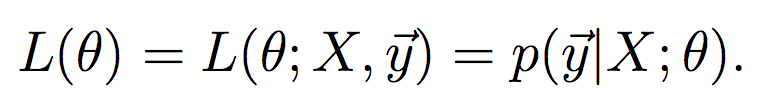
The main idea:

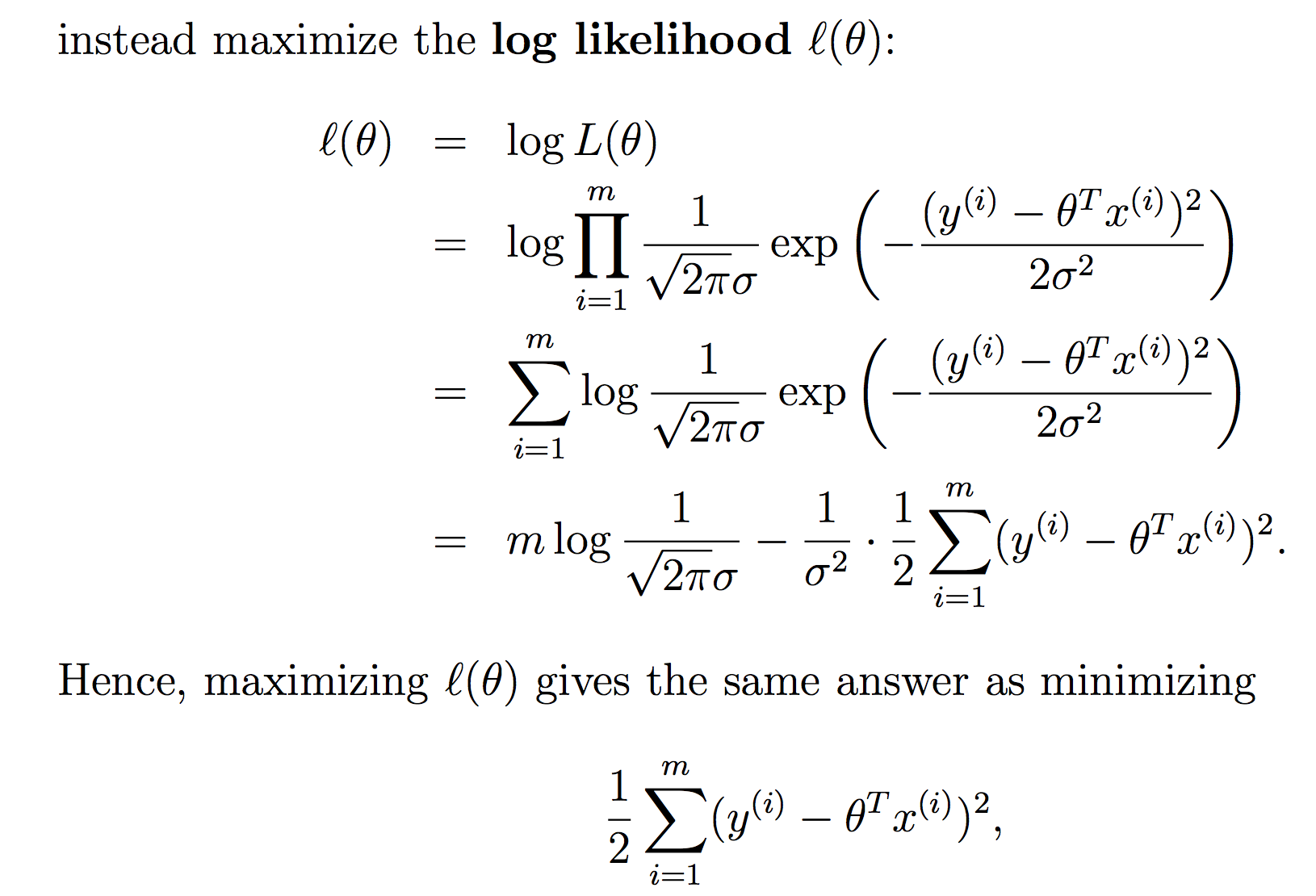
* initialize k Gaussian variables (means and variances) randomly in space.
* Compute the posterior probabilities (0<p(a)<1) for each data point (try to classify to the closest class, computes the probability the poit Xi assigned to class k).
* Re-calculate the means and the variances for every Gaussian (the calculation takes into consideration the priors).
* Repeat until converges.

How to find k?

Maximize the likelihood – the probability that the K component model will fit to all the data points. We would like the likelihood to be as large as we can.

Likelihood function:





The problem of maximizing the likelihood in equivalent to minimizing the “error” (least squares).

Occam’s razor – pick the ‘simplest’ model that fits the data.

L – likelihood – we would like to maximize

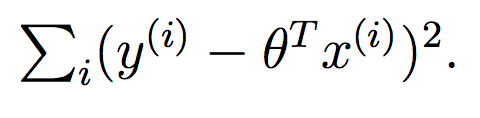
P – the number of parameters – we would like to minimize

**Parametric/non-parametric models:**

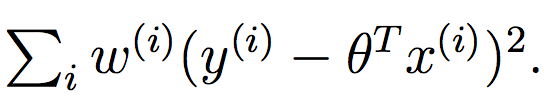
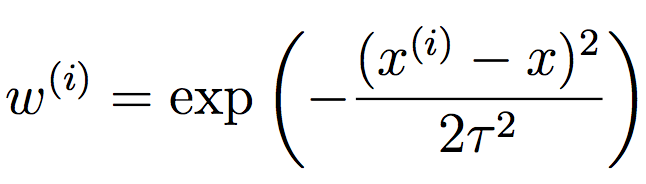
The term “non-parametric” (roughly) refers to the fact that the amount of stuff we need to keep in order to represent the hypothesis h grows linearly with the size of the training set.

A parametric model is a model which after training and finding the weights, we can forget the training data – the number of parameters/weighs is fixed and is not depend on changing number of the training data.

Parametric model: LMS least mean squares

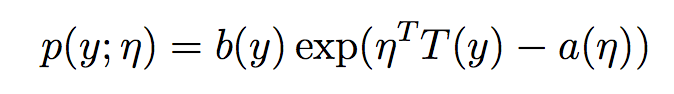


Non-parametric model: LWR weighted linear regression

**Generalized Linear Models GLM:**

General eq. for exponential family:



**Generative Learning algorithms**



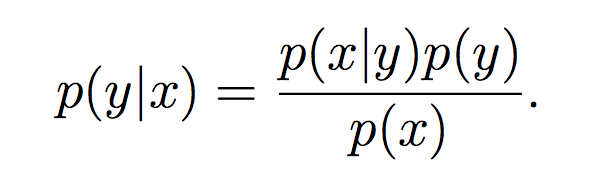
pixelRNN -

Here’s a different approach. First, looking at elephants, we can build a model of what elephants look like. Then, looking at dogs, we can build a separate model of what dogs look like. Finally, to classify a new animal, we can match the new animal against the elephant model, and match it against the dog model, to see whether the new animal looks more like the elephants or more like the dogs we had seen in the training set.

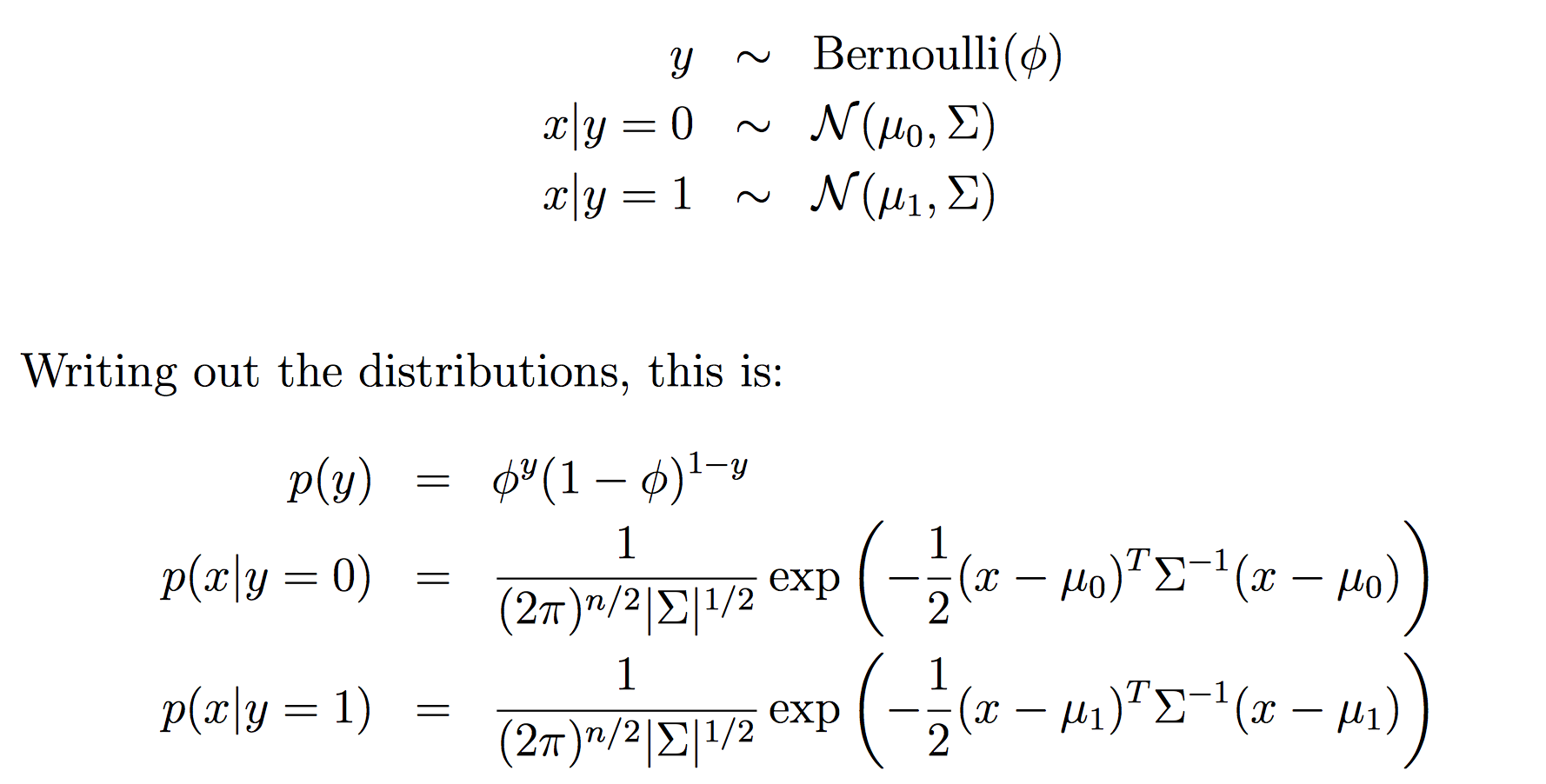
Discriminative learning algorithms - try to learn p(y|x) directly or algorithms that try to learn mappings directly from the space of inputs X .

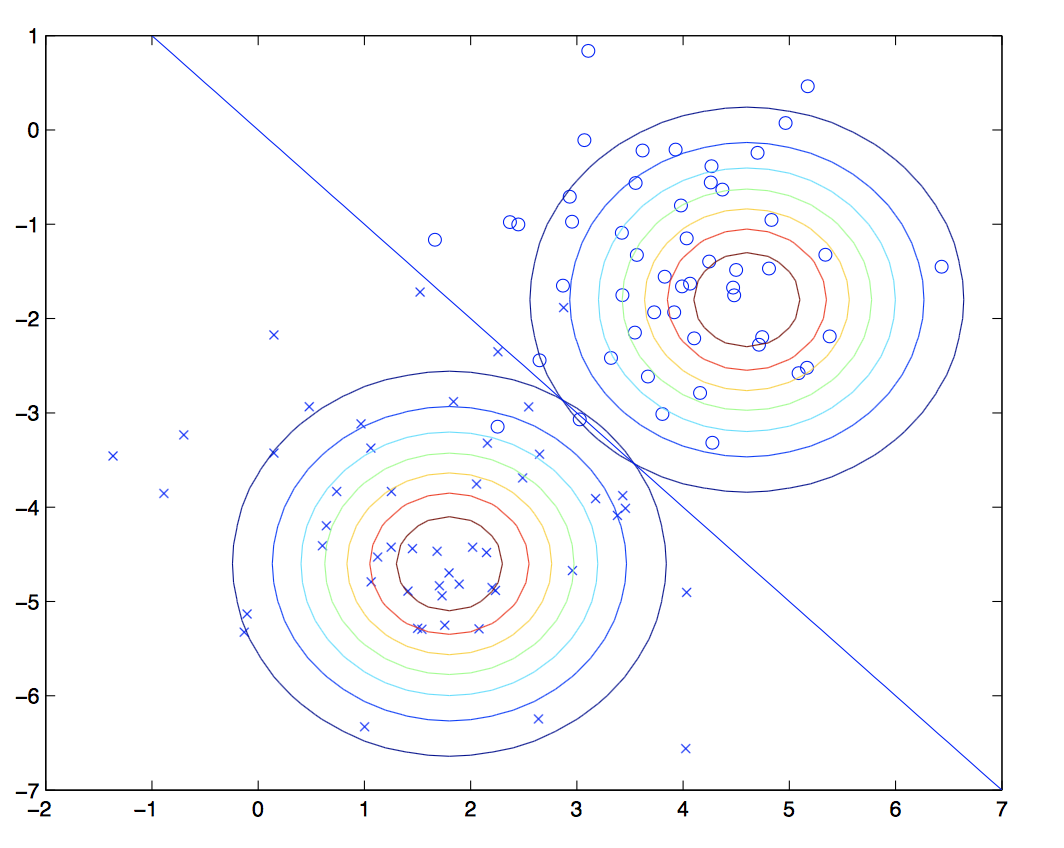
generative learning algorithms - try to model p(x|y). generate a general model of each class and trying to classify the current example class.

Bayes rule:



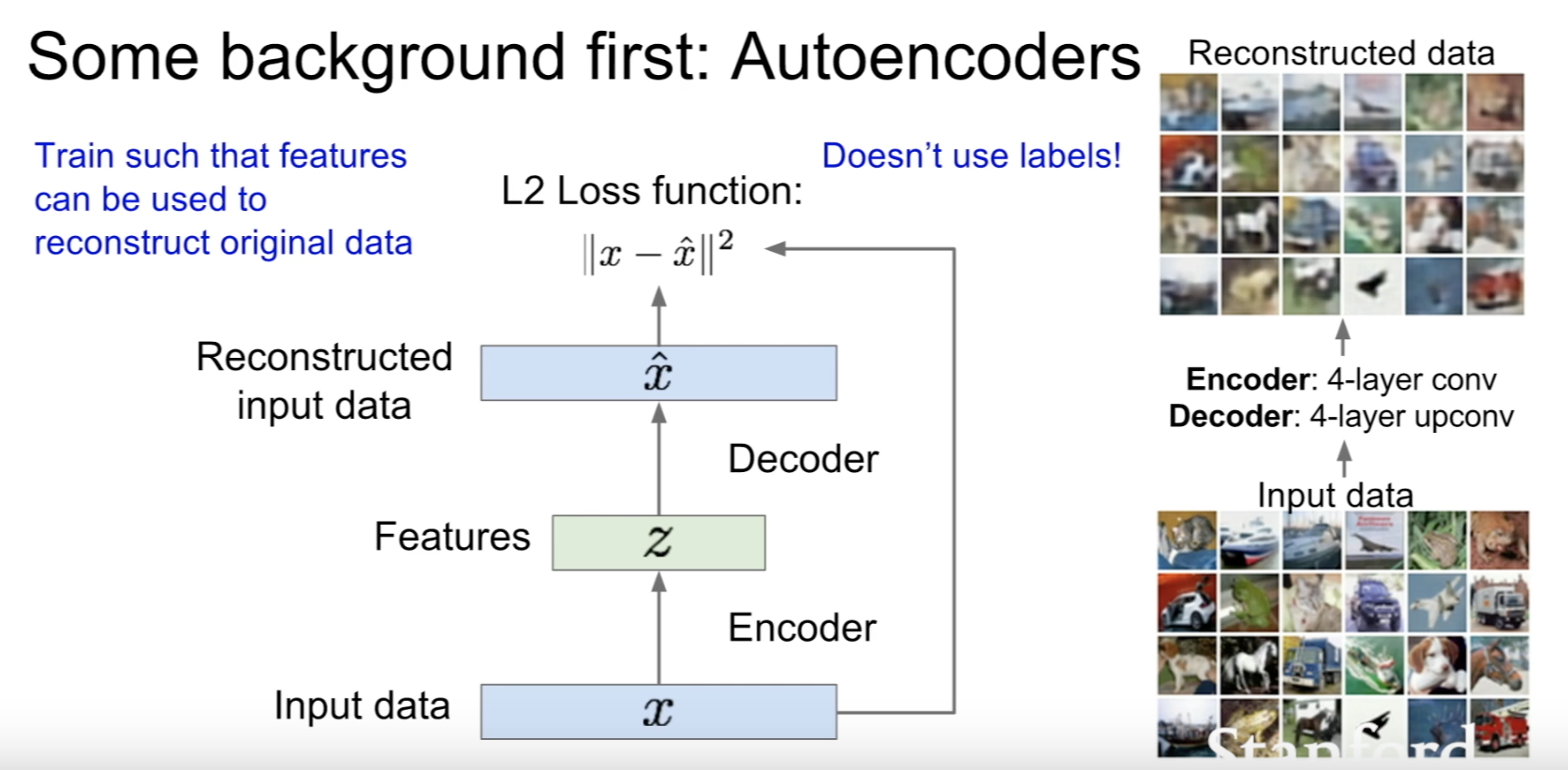
Gaussian discriminant analysis (GDA)





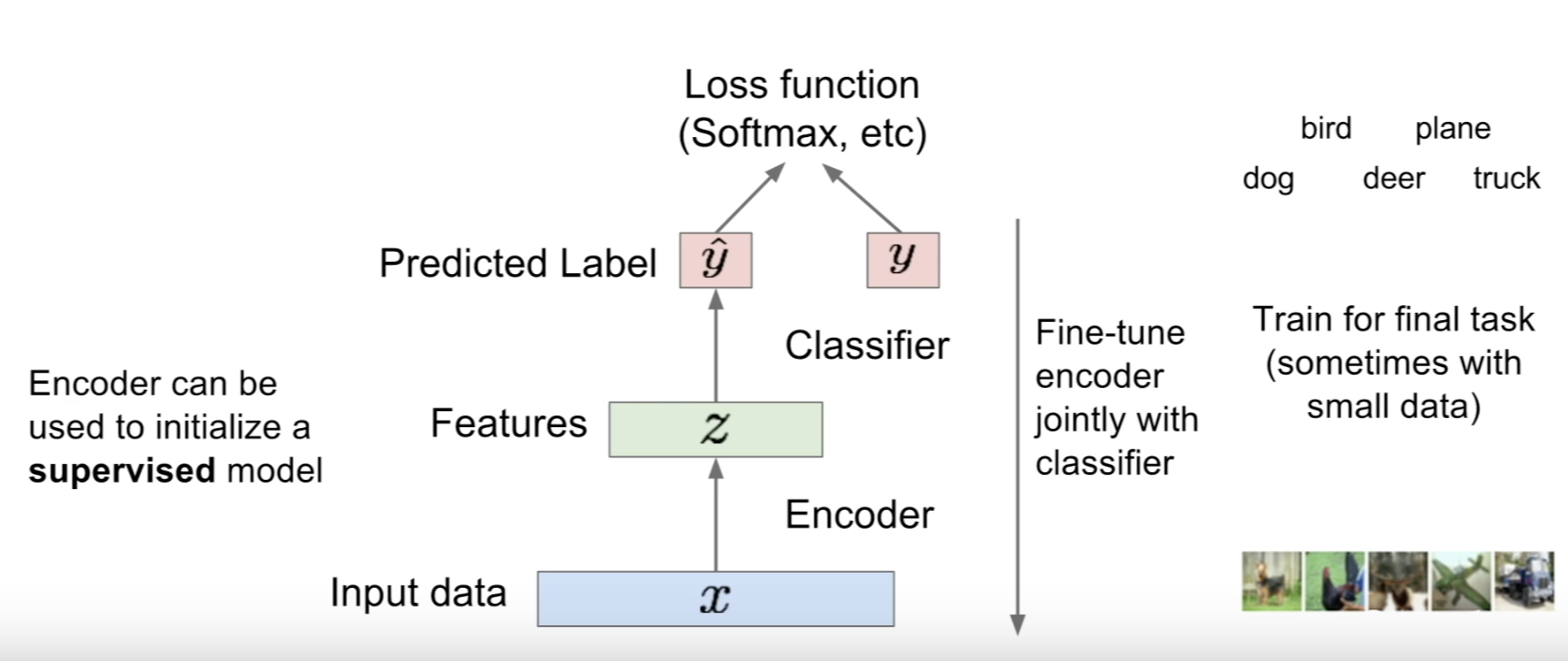
builds a model of Gaussian’s that classifies the data distributions into classes.

**Autoencoders:**



unsupervised learning - feature vector Z in used of reconstruction of the input data (with spatial context).

After learning, we will throw the decoder and add a classfier, we will use the network now for supervised learning and classification.



**Naive Bayes**

To model p(x|y), we will therefore make a very strong assumption. We will assume that the Xi’s are conditionally independent given y.

**Mixtures of Gaussians and the EM algorithm**

**Graphical modules**

\*taken from Stanford lectures.

Introduction: <https://www.youtube.com/watch?v=6AVurePzK3Y>

Markov networks: <https://www.youtube.com/watch?v=SH1K4RtX9uQ>

<https://www.youtube.com/watch?v=WPSQfOkb1M8&list=PL50E6E80E8525B59C>

<https://www.youtube.com/watch?v=ju1Grt2hdko>

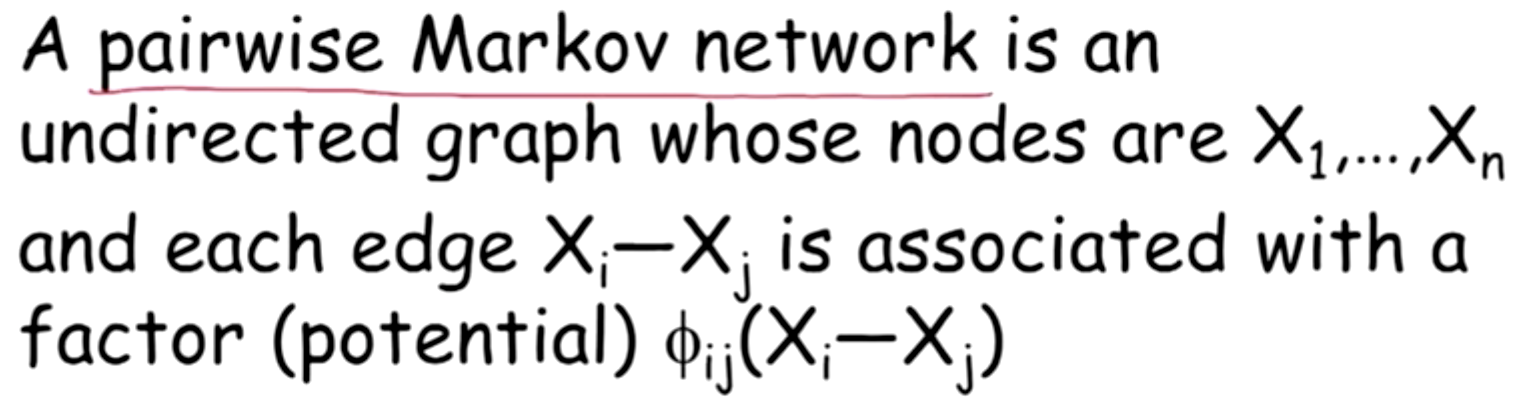
short summery in my words:

Bayesian networks – defines the relations between random variables as directed graph.

Markov networks – defines the relations between random variables as an undirected graph.

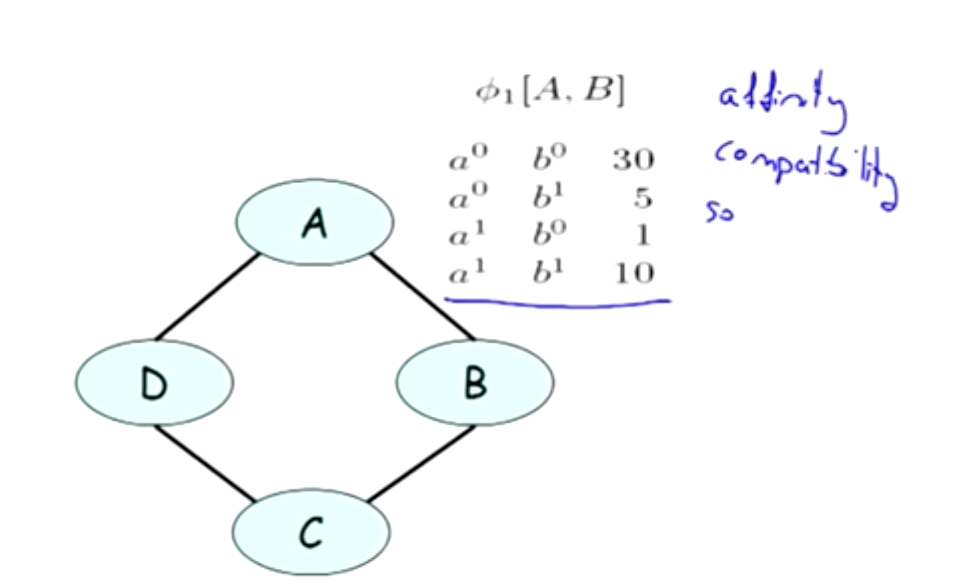
In image segmentation task, the random variables are the labels of the superpixels.

Focus on MRF:



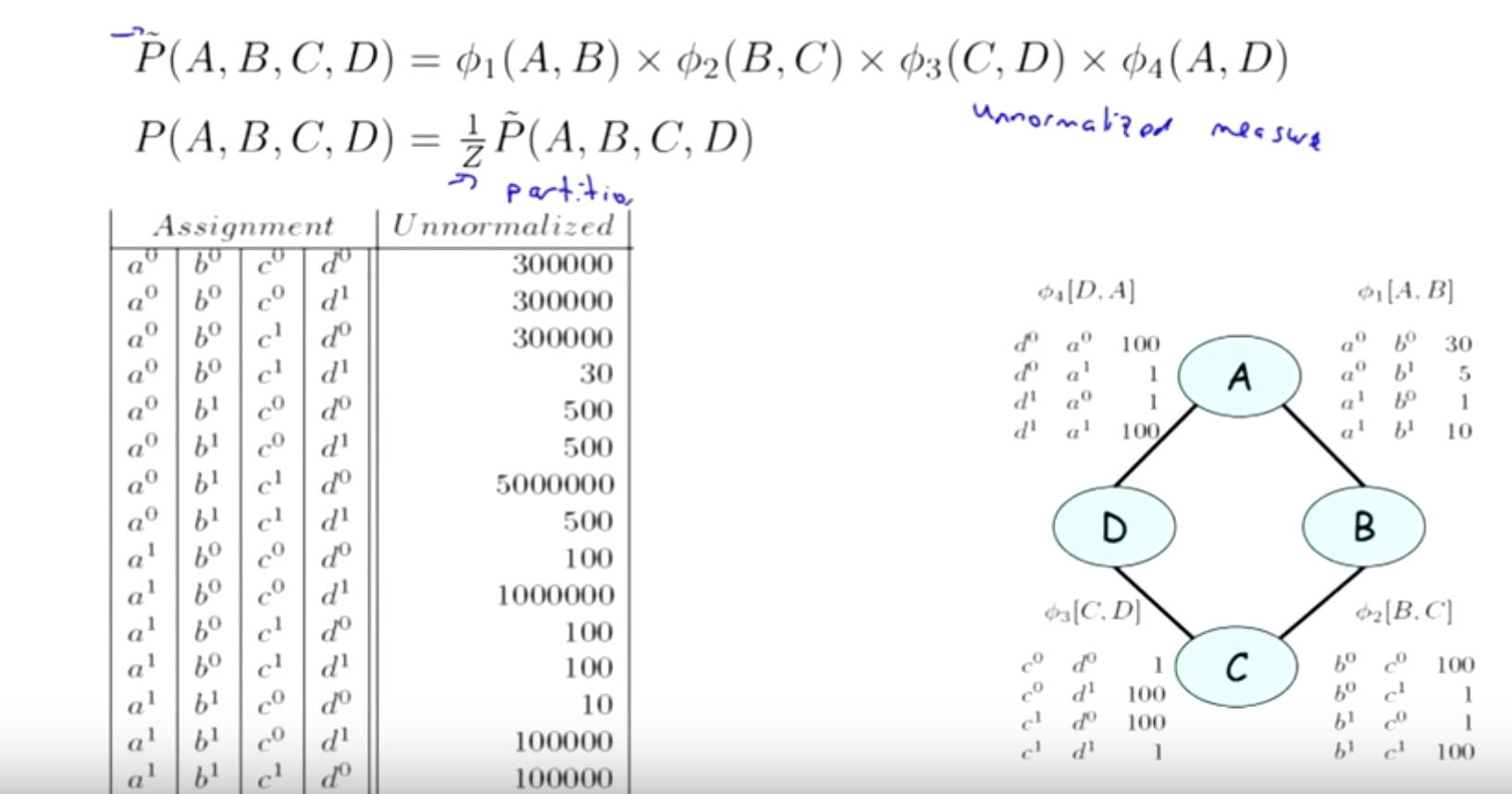
Toy data example of MRF:

MRF of an image will be  **a grid.**



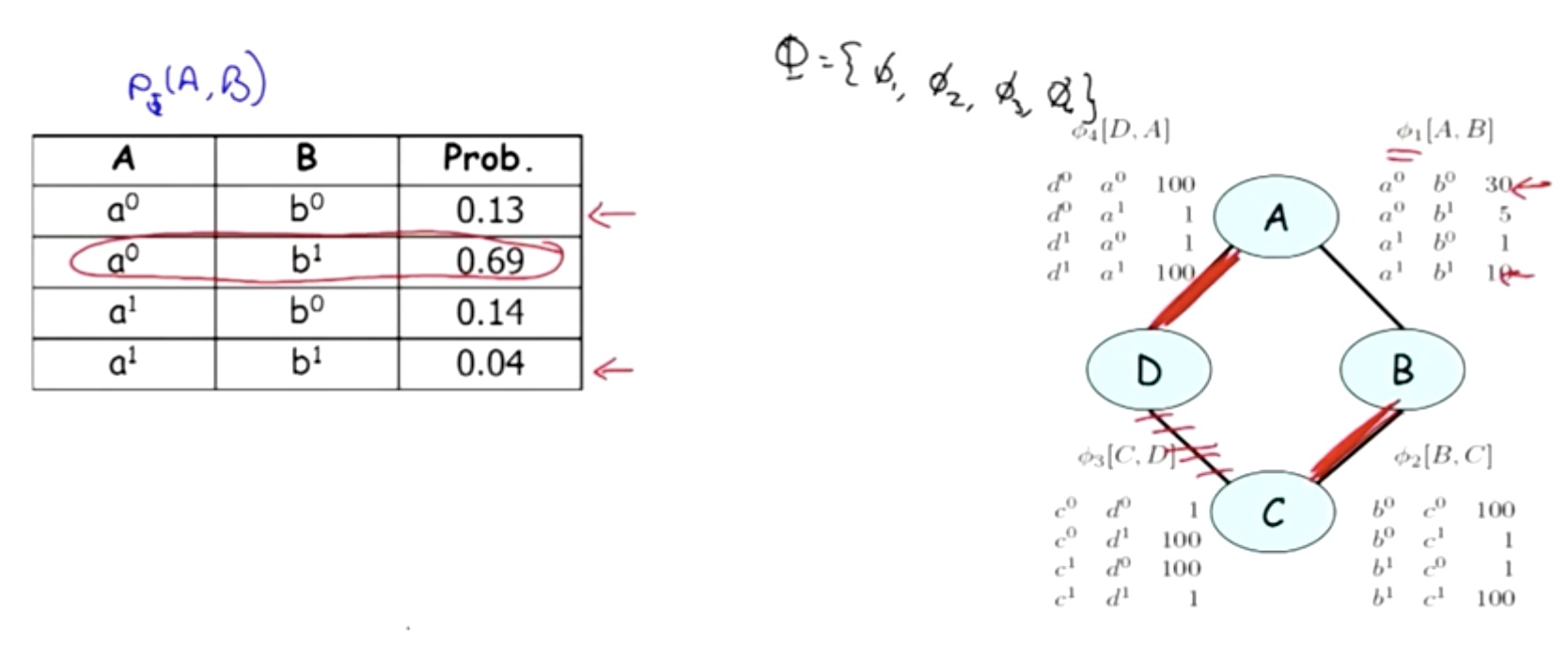
the random variables are the nodes, the relations between the nodes (arches) are describes as affinity functions /compatibility functions/soft constrains. This function stands for the cost/weight of the pair.

We would like to calculate the probability of a path given by: a-b-c-d. To do so we will define the following measures:

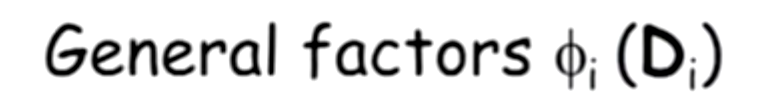


1/Z – normalized constant to get a distribution.

There is no direct connection between the arches factors and the probability distribution of the arche, as can be seen here:

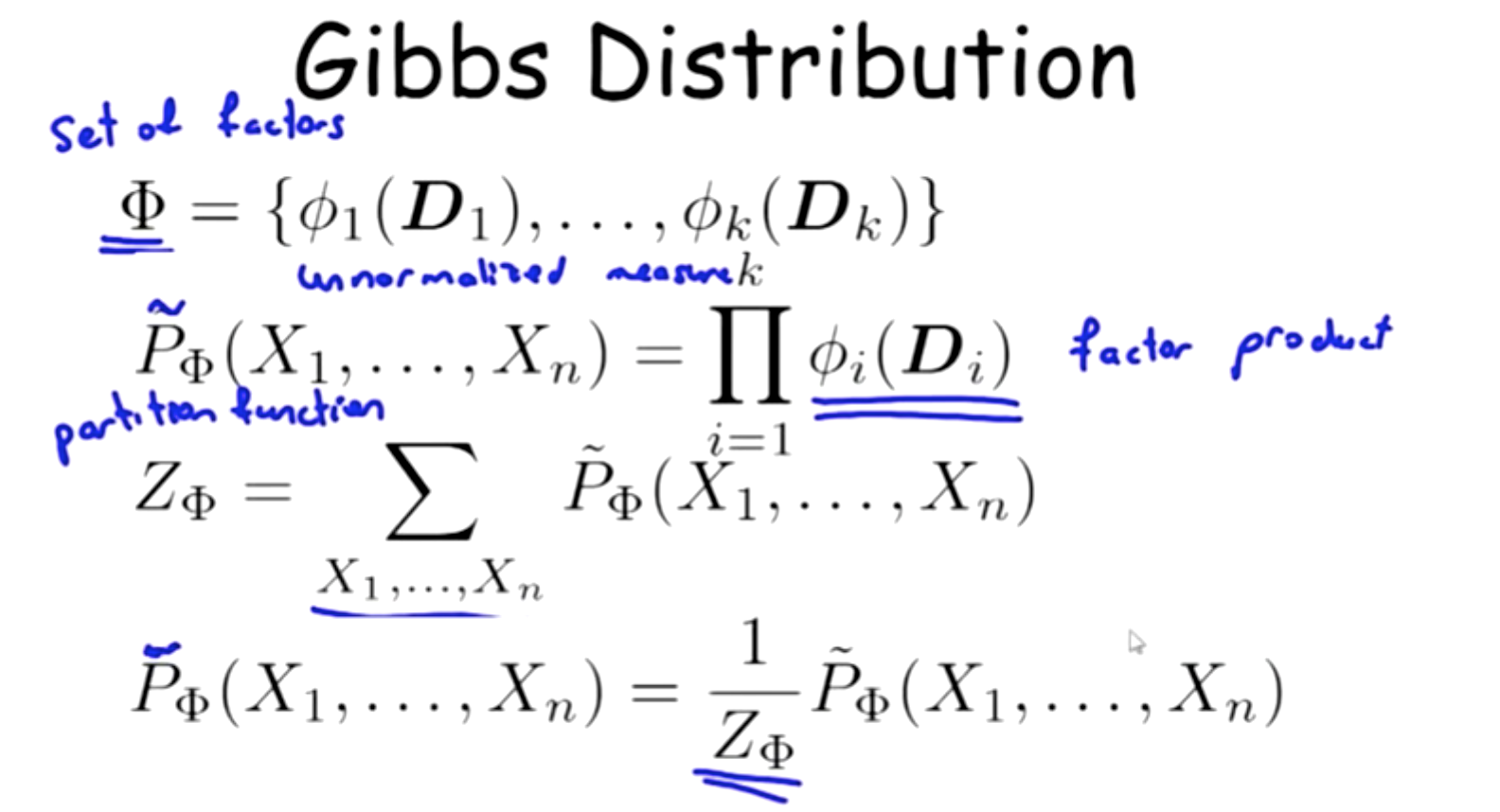


General Gibbs Distribution

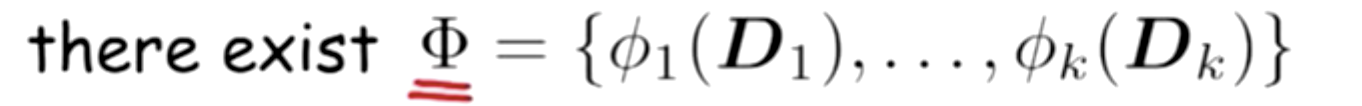


PHI is a general factor - value for a Di combination of voxels in the graph.

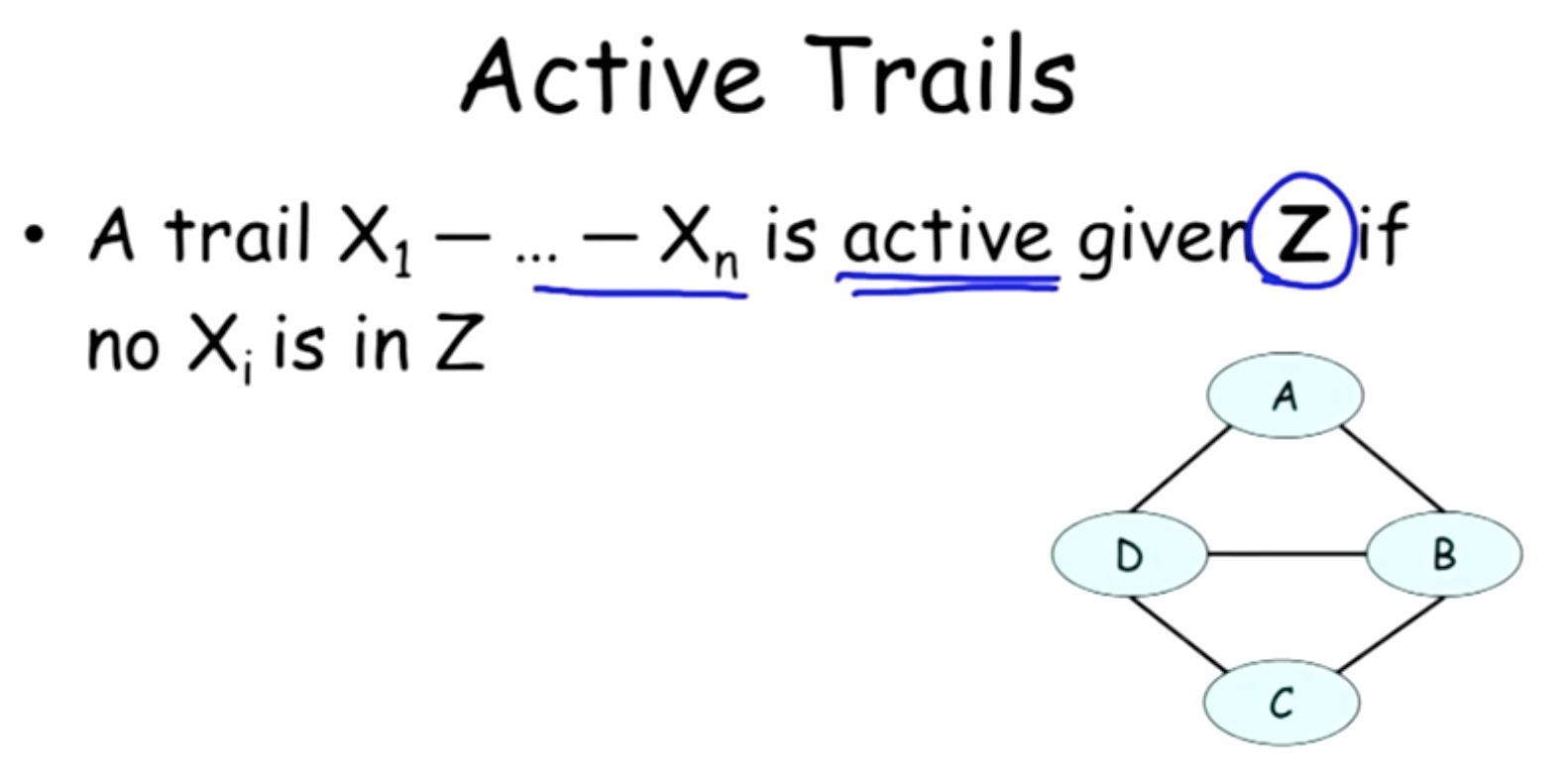
H phi – the induced Markov network, it means that there is as edge for every [Xi,Xj] where Xi,Xj are in phi.



factorization – we are looking for distribution P as: P=Pphi such that:

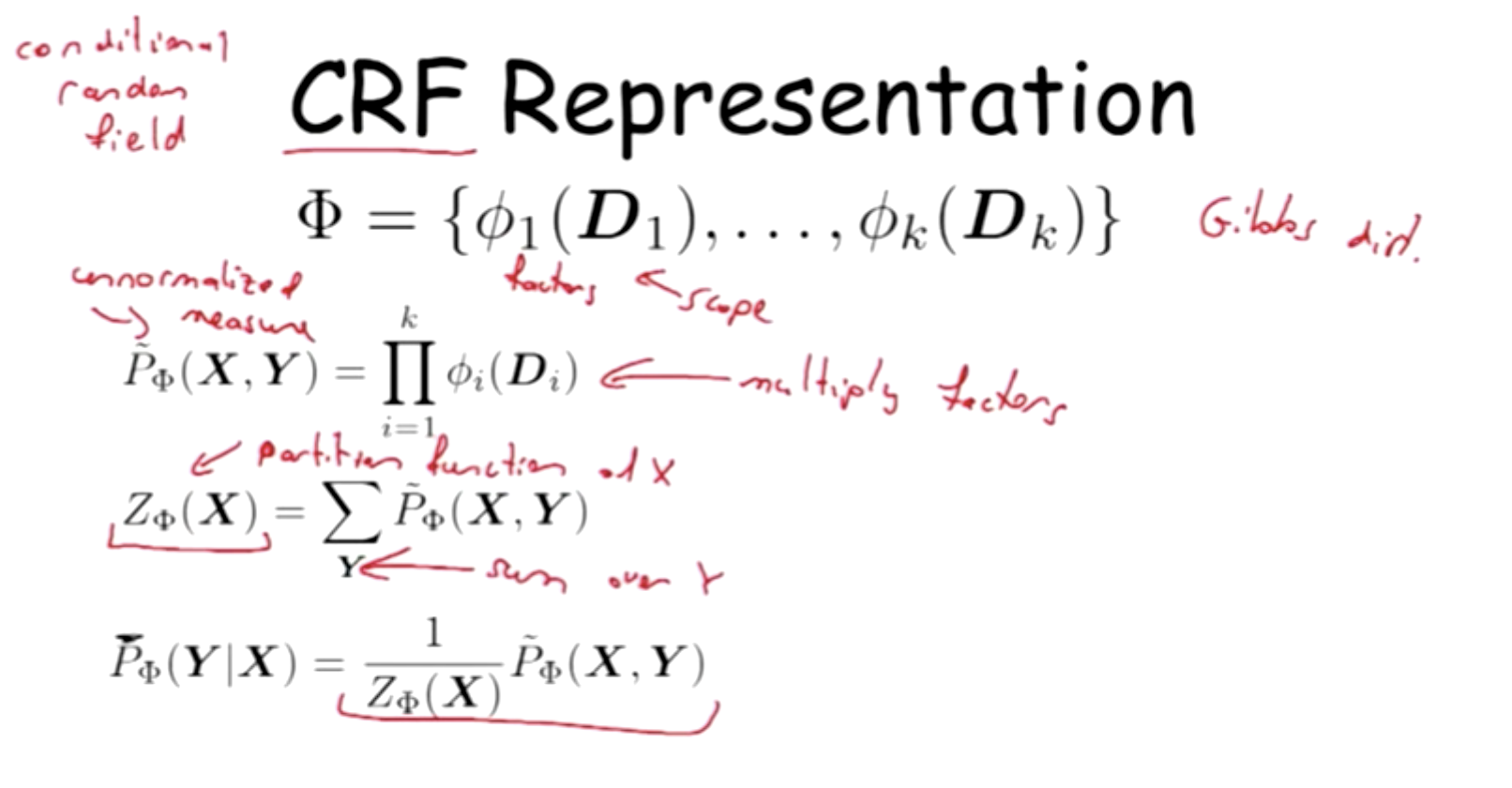


and H is the induced graph for phi. (looking for graph that it’s normalized measure is the desire P).

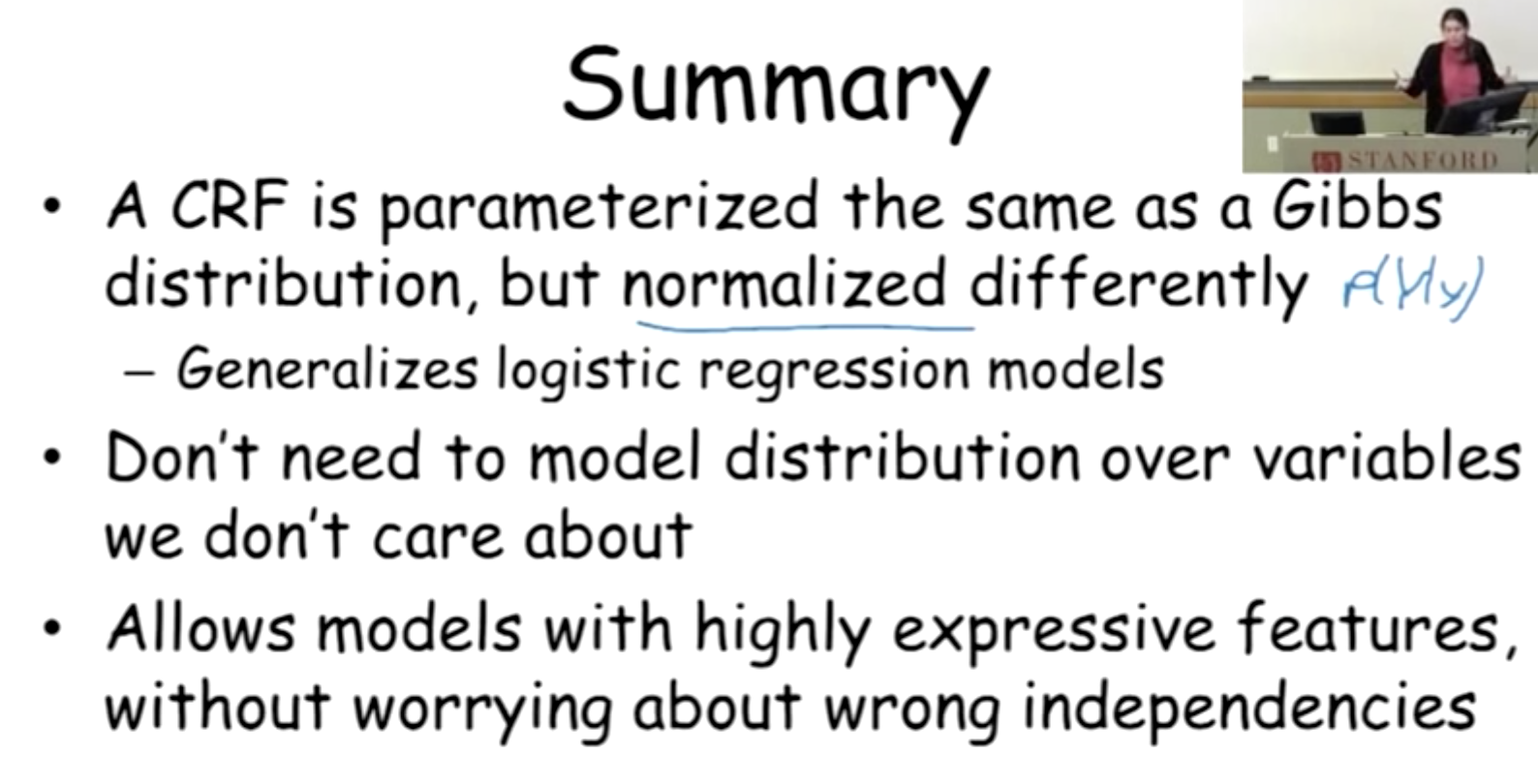


Conditional Random Fields

We are trying to find P(y/x) so we don’t care about correlations and relations between Xi’s in the graph -> function of X.



  models how all the X’s come together to influence the probability of y.



**Architecture ideas for the final project:**

* CNN & RFC (maybe a modification of DeepMedic)
* ReSeg / use of RNN for spatial dependencies (3D) - (CNN&RNN) – extension to 3D for contextual information
* 3D-patch CNN
* SegNet – encoding/decoding , new decoding method

Classic segmentation project:

* To check GMM with/without tumor
* Feature extraction for RF
* To apply the network framework

DeepMedic

* Ansambe of NN
* Post-processing of MRF
* Resideaul
* MCMC – sample from an unknown distribiotiong
* Feature extraction like GMM in pre-processing process

Less data in transfer learning, less layer training

ReSeg

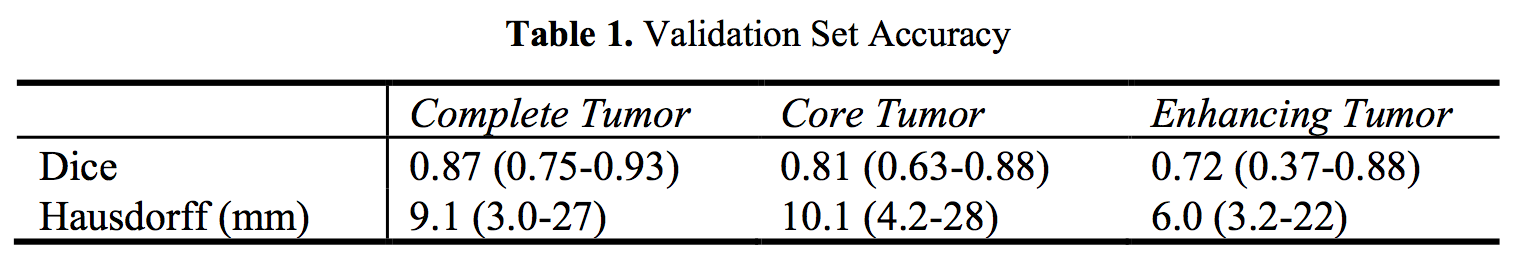
* Ensamble of patches in different sizes. For each track we will train RNN. At the end few layers to sum all the data.

**CNN ideas:**

1. **DAG-CNN**

[1] Fully Convolutional Neural Networks with Hyperlocal Features for Brain Tumor Segmentation

* Yang S, Ramanan D. Multi-Scale Recognition With DAG-CNNs. 2015. p. 1215–23.
* P.BaldiandG.Pollastri.Theprincipleddesignoflarge-scale recursive neural network architectures–dag-rnns and the pro- tein structure prediction problem. JMLR, 2003. 2

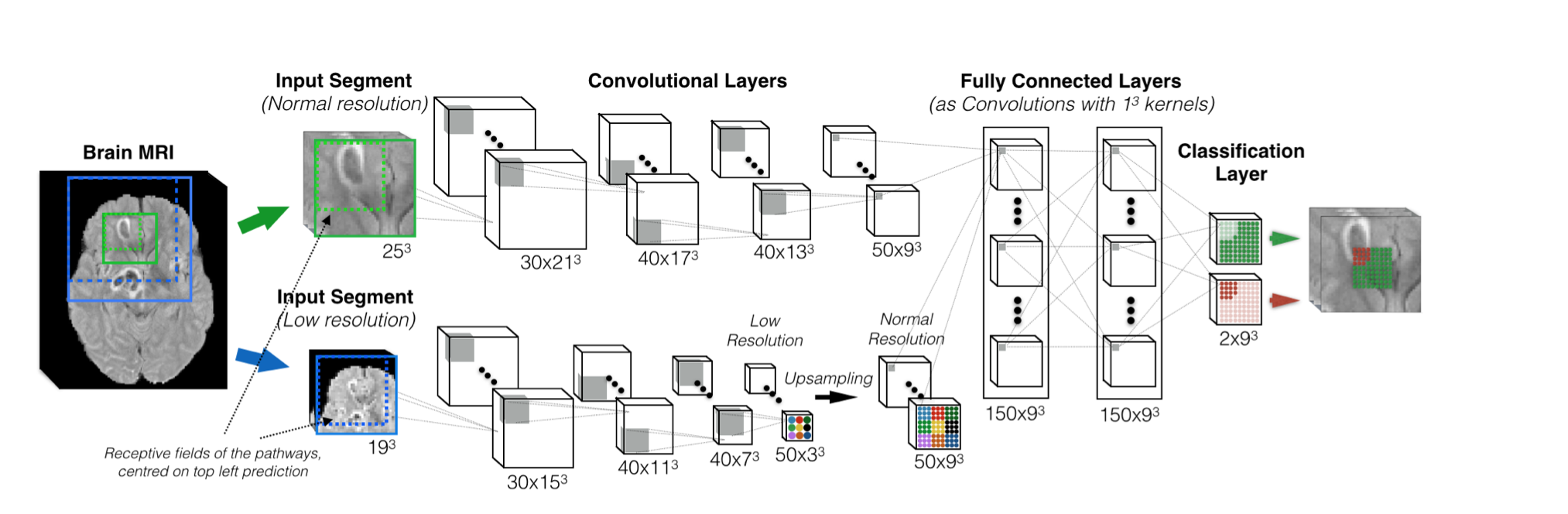


This hyperlocal quality of medical image segmentation can be cap- tured with a direct-acyclic graph (DAG-CNN) architecture that re-introduces the orig- inal input image into the network just two layers prior to final classification

[2] Nabla-net: a deep dag-like convolutional architecture for biomedical image segmentation: application to high- and low-grade glioma segmentation

Nabla net is a dag-like deep neural network architecture, combining a fully-convolutional pathway learning low-level features and an encoder-decoder network learning high-level features

1. **DeepMedic**

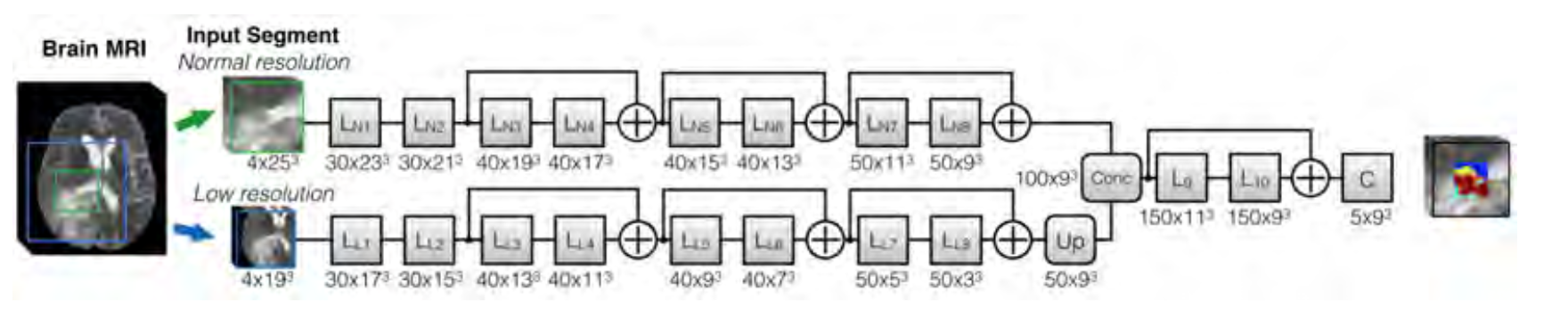
****

* 3D CNN
* 11 layers deep
* 3D CNN that produces highly accurate, soft segmentation maps
* Multi-Scale Processing via Parallel Convolutional Pathways
* Fully connected 3D CRF that imposes regularization constraints
* two articles in BARTS 2016 used this method:
  + Extend to residual connections in this article: DeepMedic on Brain Tumor Segmentation
  + 3D Convolutional Networks for Brain Tumor Segmentation (offered 3 models, one of them is based on DeepMedic)

**TODO:**

1. Kamnitsas,K.,Ledig,C.,Newcombe,V.F.,Simpson,J.P.,Kane,A.D.,Menon,D.K.,Rueckert, D., Glocker, B.: Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. arXiv preprint arXiv:1603.05959 (2016)
2. Kra ̈henbu ̈hl, P., Koltun, V., 2011. Efficient inference in fully connected CRFs with gaussian edge potentials. Adv. Neural Inf. Process. Syst.
3. Urban, G., Bendszus, M., Hamprecht, F., Kleesiek, J.: Multi-modal brain tumor segmentation using deep convolutional neural networks. in proc of BRATS-MICCAI (2014)
4. Pereira, S., Pinto, A., Alves, V., Silva, C.A.: Brain tumor segmentation using convolutional neural networks in mri images. IEEE transactions on medical imaging 35(5) (2016) 1240–1251
5. akas, S., Zeng, K., Sotiras, A., Rathore, S., Akbari, H., Gaonkar, B., Rozycki, M., Pati, S., Davatzikos, C.: Glistrboost: Combining multimodal mri segmentation, registration, and biophysical tumor growth modeling with gradient boosting machines for glioma segmentation. In: International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic  Brain Injuries, Springer (2015) 144–155
6. Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin,  P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. arXiv preprint  arXiv:1505.03540 (2015)
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recog-  nition. arXiv preprint arXiv:1409.1556 (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv  preprint arXiv:1512.03385 (2015)

DeepMedic on Brain Tumor Segmentation



1. Kamnitsas,K.,Ledig,C.,Newcombe,V.F.,Simpson,J.P.,Kane,A.D.,Menon,D.K.,Rueckert, D., Glocker, B.: Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. arXiv preprint arXiv:1603.05959 (2016)
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recog-  nition. arXiv preprint arXiv:1409.1556 (2014)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv  preprint arXiv:1512.03385 (2015)

**SegNet: Encoder-Decoder Architecture**

The challenge is to choose the right decoder, two methods have propused:

SegNet may be better.

SegNet - Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder- decoder architecture for image segmentation. arxiv:1511.00561 (2015)

FCN - Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. arXiv preprint arXiv:1411.4038 (2014)

Future thought, maybe LSTM.

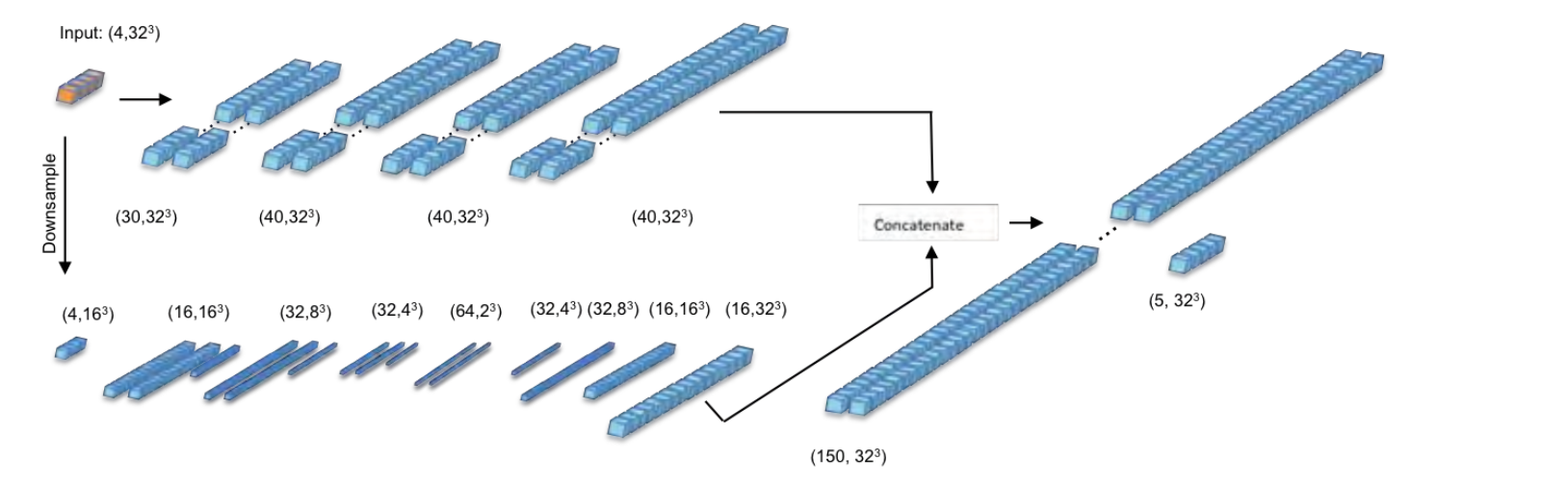
Nabla net is a dag-like deep neural network architecture, combining a fully-convolutional pathway learning low-level features and an encoder-decoder network learning high-level features

3D Convolutional Networks for Brain Tumor Segmentation

3D CNN networks, this work examined the advantages of 3D CNN.

3 models:

Deepmedic 3D, 3DNET1, 3DNET2



Improving segment boundary classification for Brain Tumor Segmentation and longitudinal disease progression

\* two stages training for fine training.

\* more weight to the edges pixels for approves accuracy

Brain tumor segmentation using a fully convolutional neural network with conditional random fields

**Conditional Random Fields (CRF),**

CRF: CRF could be formulated as a recurrent neural network (RNN), re- ferred to CRF-RNN [4], making it possible to integrate a FCNN and a CRF network as one deep network and train it using a typical back-propagation al- gorithm.

S. Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes et al. Conditional Ran- dom Fields as Recurrent Neural Networks, ICCV 2015, pp. 1529-1537.

**ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation**

free-style idea:

\*Transpose convolution – an upsampling learning method, input small image gives weight to a multipicaion with a slidding filter. At the output we sum all the overlaps values.

\*2D slices vs 3D patches ?

\*3D CNN

\* find tumor ROI and then fine tuning segmentation for regions segmentation.

\* batch normalization for learning

\* Markov or Conditional Random Fields (CRFs) have become one of the most widely-used graphical models in image understanding [10]

ARTICLES:

1. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomed-  ical Image Segmentation. arXiv:1505.04597v1 (2015)
2. Noh, H., Hong, S., Han B.: Learning Deconvolution Network for Semantic Segmen-  tation. ICCV, Santiago, Chile (2015)
3. Long, J., Shelharmer, E., Darrel, T.: Fully Convolutional Networks for Semantic  Segmentation. CVPR, Boston, USA (2015)
4. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional  Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561 (2015)

“classic” segmentation ideas:

* ROI and RF
* Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)