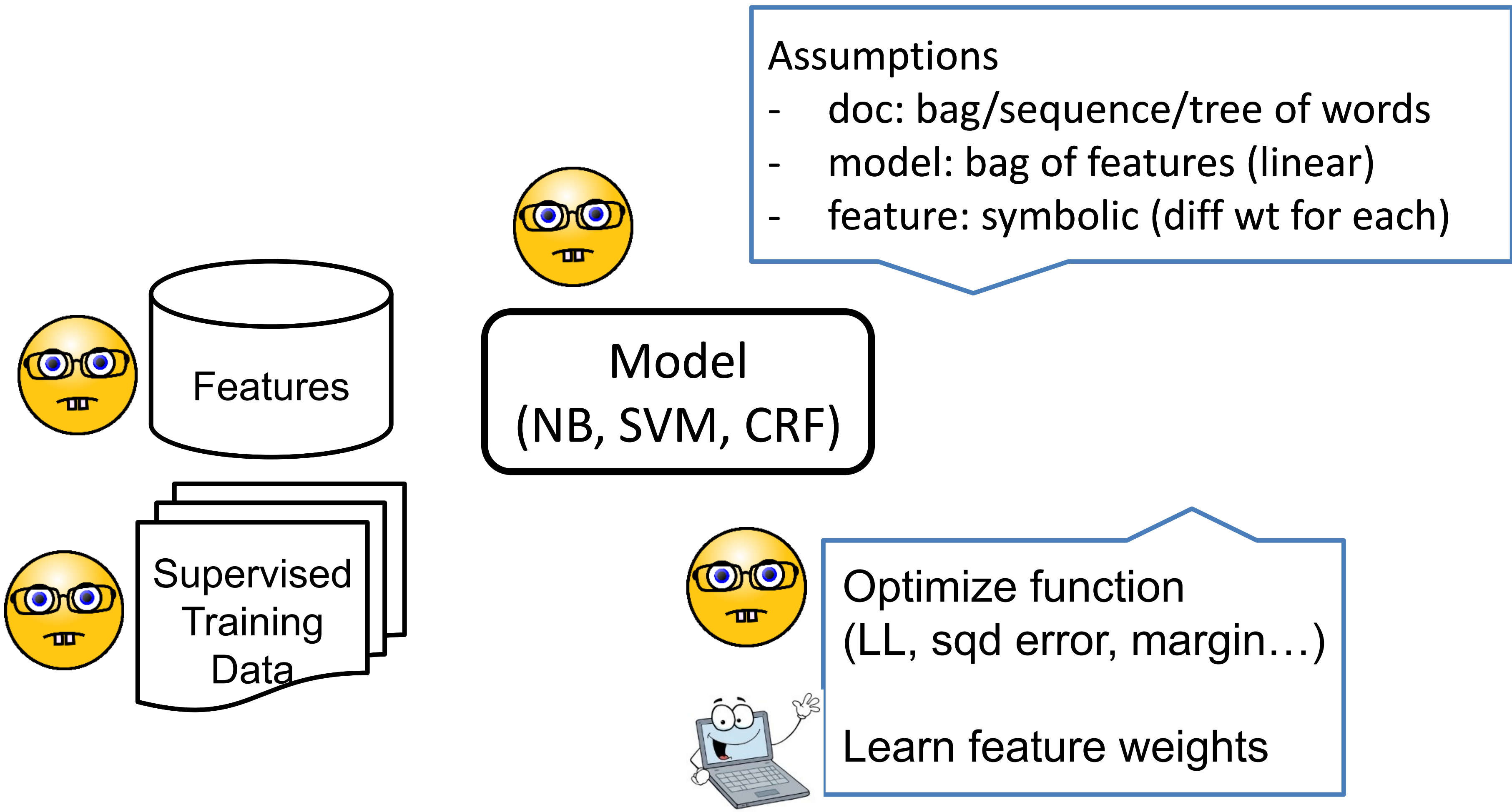
An Intro to Deep Learning for NLP

Mausam

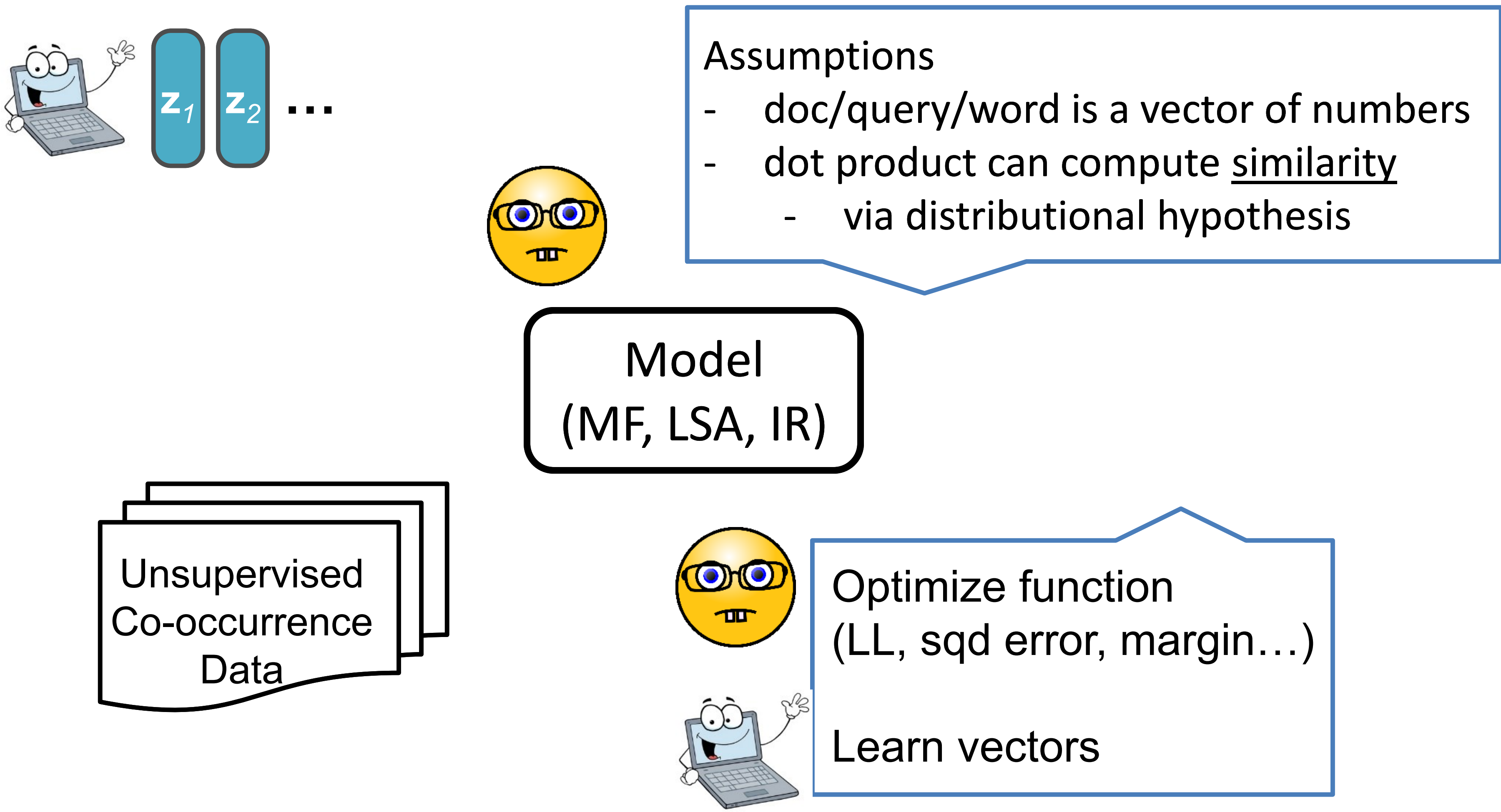
Disclaimer: this is an outsider's understanding. Some details may be inaccurate

(several slides by Yoav Goldberg & Graham Neubig)

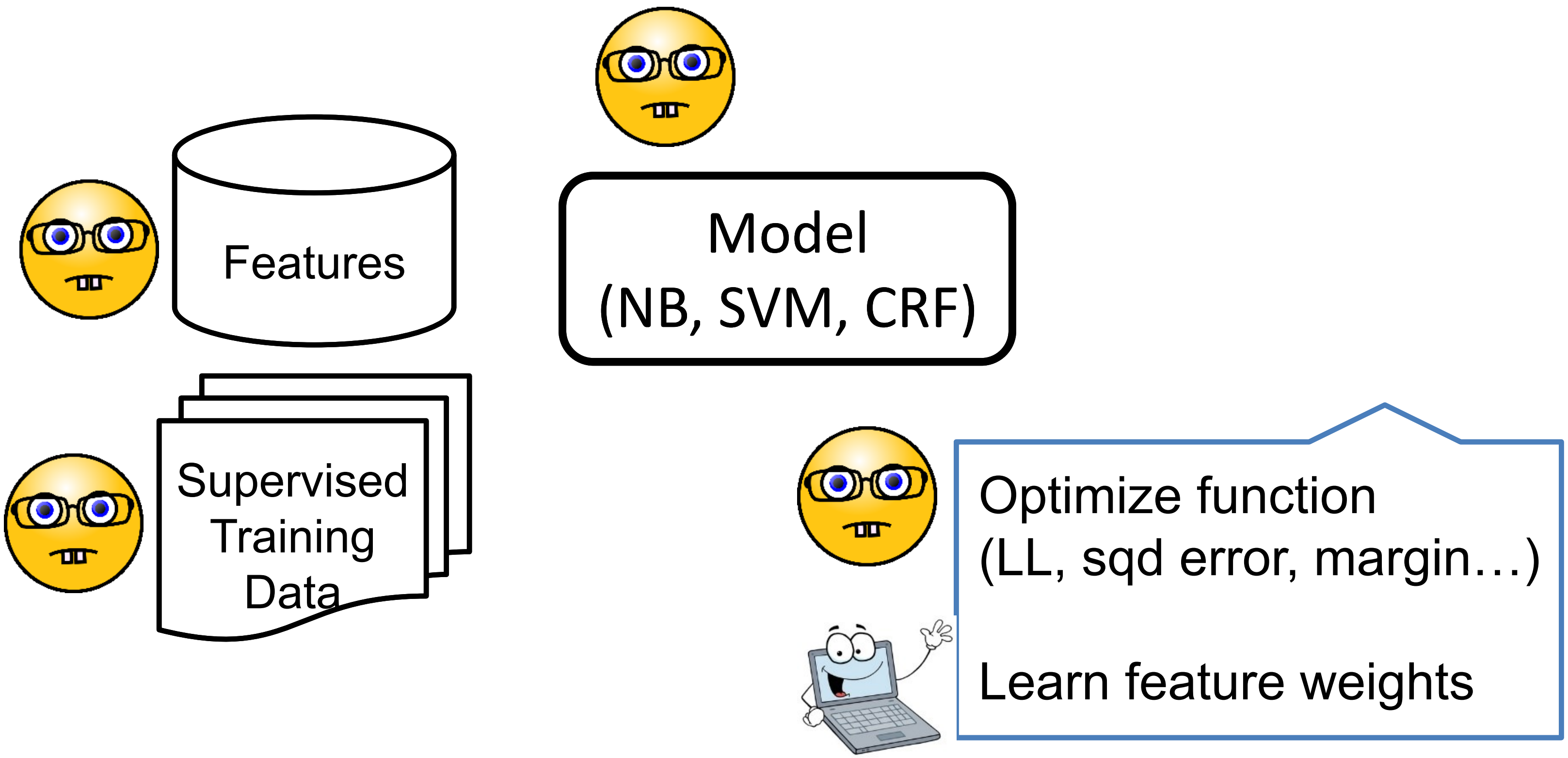
NLP before DL #1



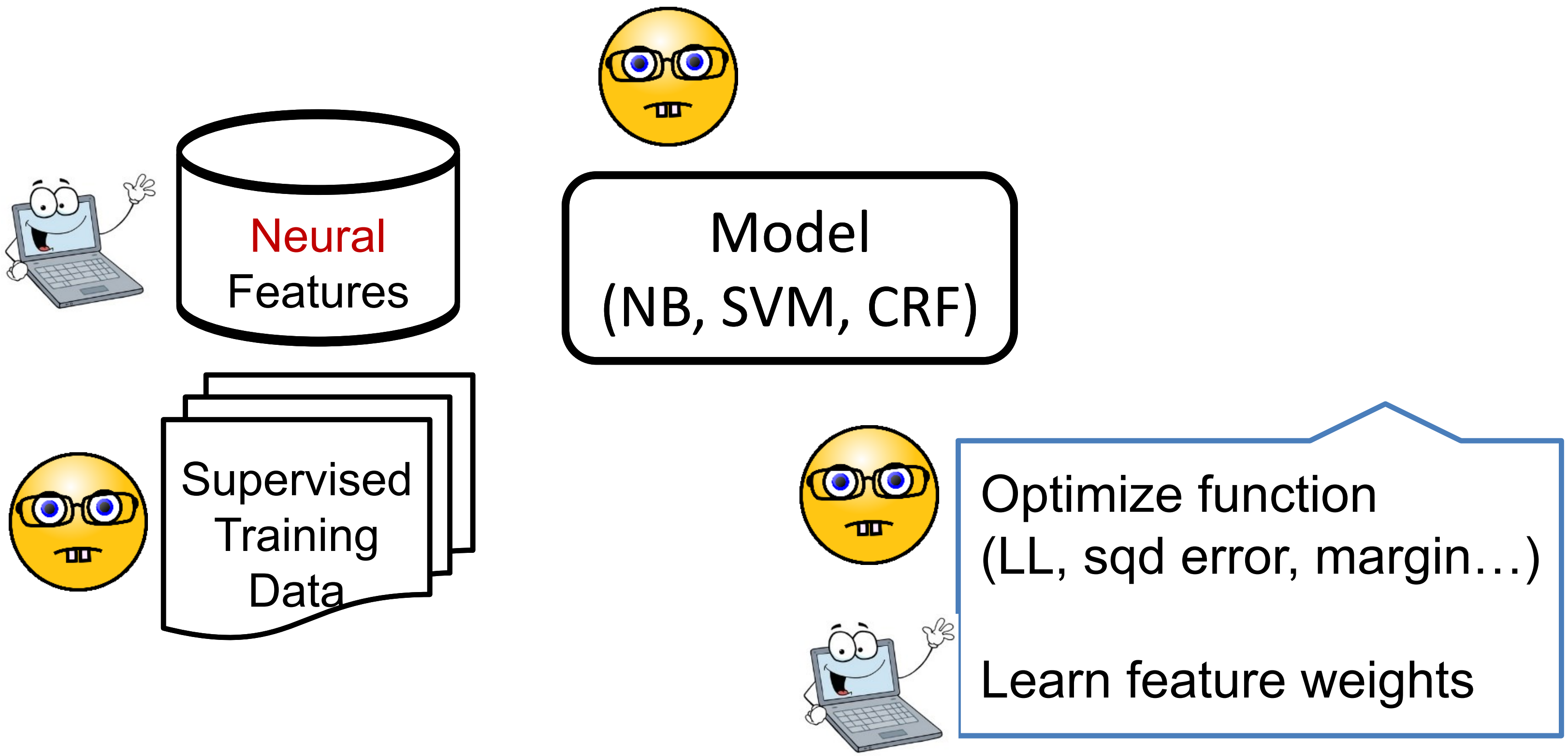
NLP before DL #2



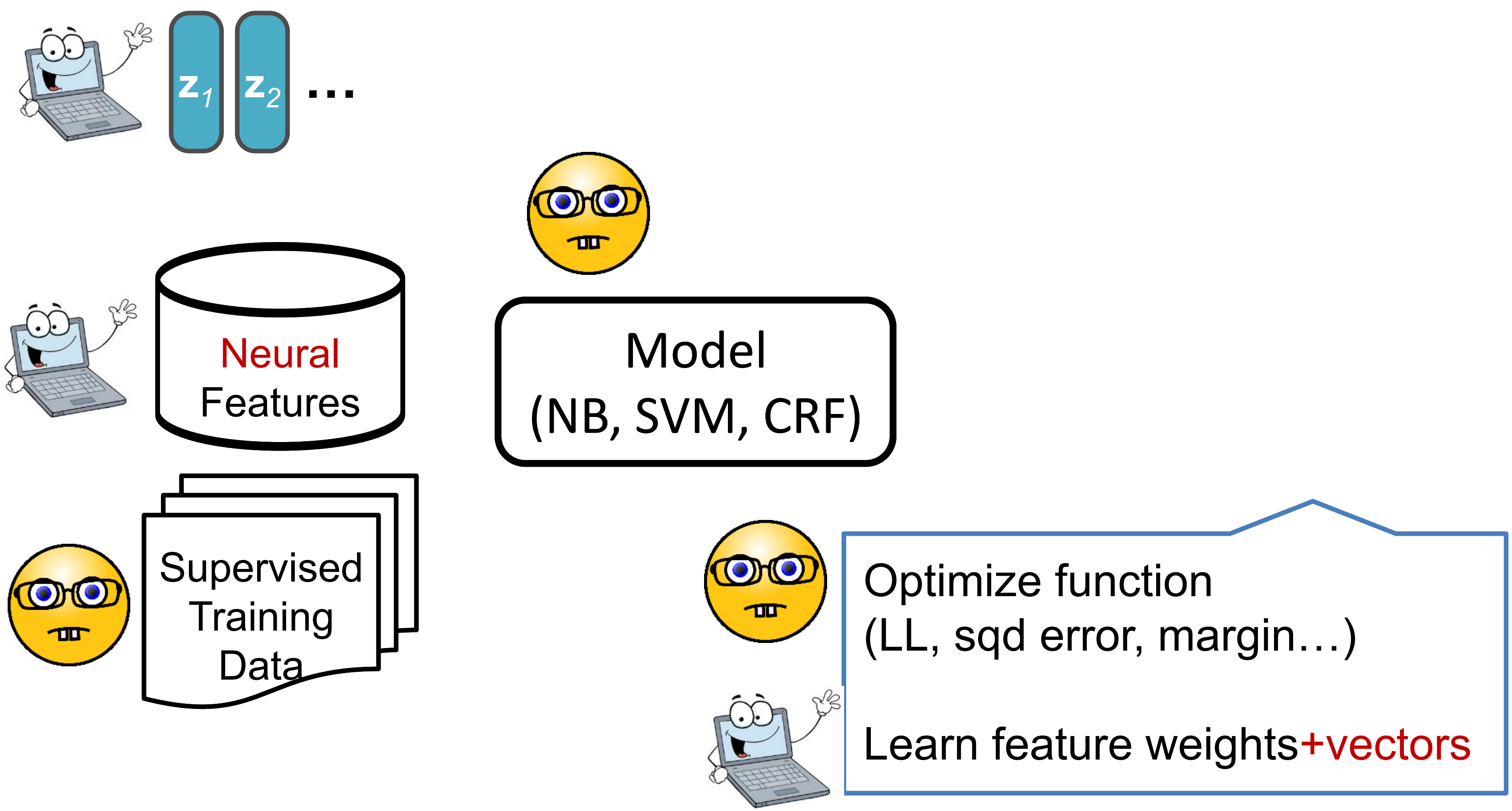
NLP with DL


Features
Supervised Training Data
Model
(NB, SVM, CRF)
Optimize function
(LL, sqd error, margin…)
Learn feature weights

NLP with DL

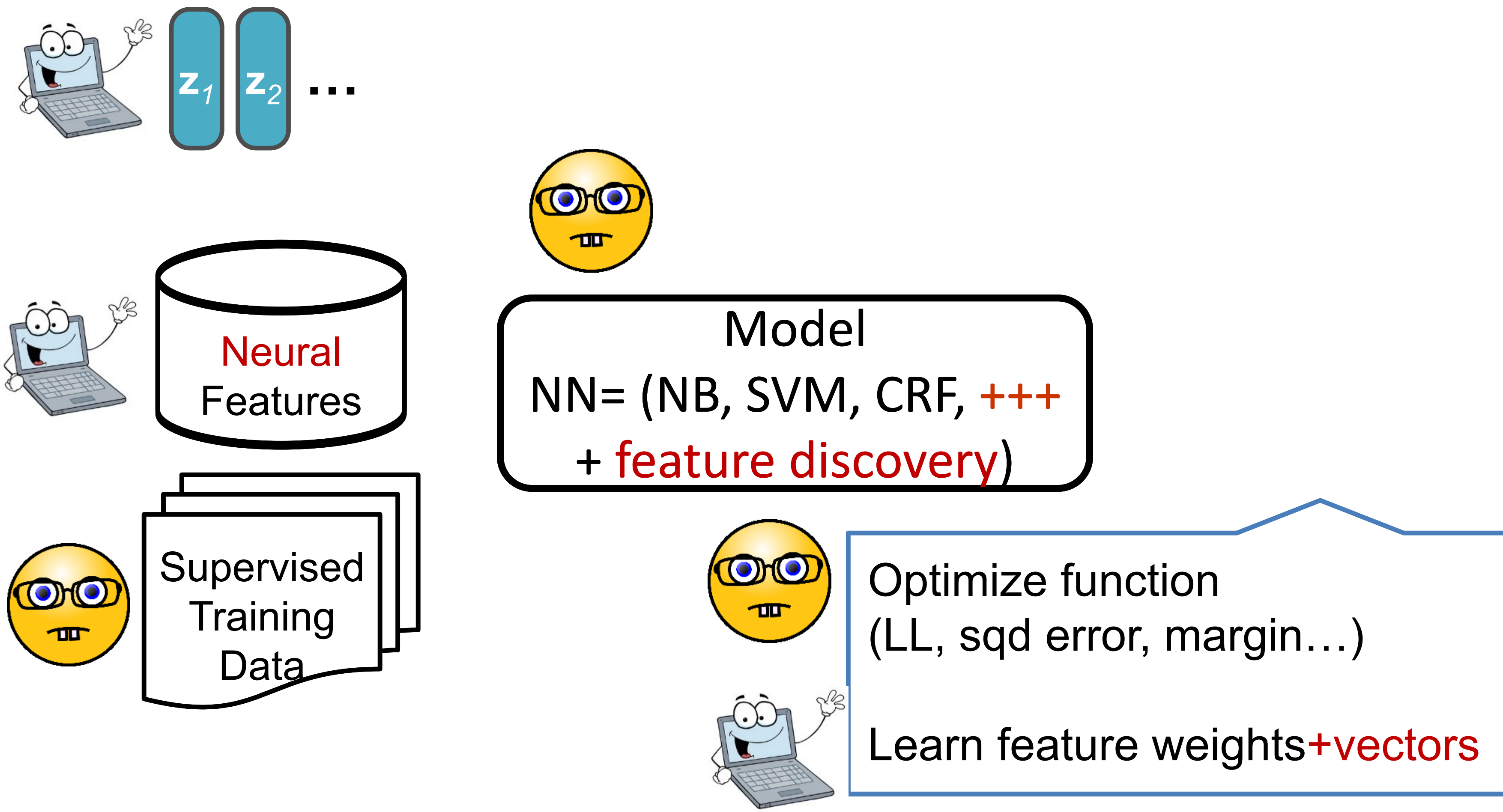

Neural
Features
Model
(NB, SVM, CRF)
Supervised
Training
Data
Optimize function
(LL, sqd error, margin…)
Learn feature weights

NLP with DL



z_1 z_2 …
Neural
Features
Supervised
Training
Data
Model
(NB, SVM, CRF)
Optimize function
(LL, sqd error, margin…)
Learn feature weights+vectors

NLP with DL



z_1 z_2 …
Neural
Features
Supervised
Training
Data
Model
NN= (NB, SVM, CRF, +++
+ feature discovery)
Optimize function
(LL, sqd error, margin…)
Learn feature weights+vectors

NLP with DL

z_1 z_2 …

Neural Features

Supervised Training Data

Model

NN= (NB, SVM, CRF, +++

+ feature discovery)

Assumptions

- doc/query/word is a vector of numbers
- doc: bag/sequence/tree of words
- feature: neural (weights are shared)
- model: bag/seq of features (non-linear)

Optimize function
(LL, sqd error, margin…)

Learn feature weights+vectors

Meta-thoughts

Features

- Learned
- in a task specific end2end way
- not limited by human creativity

Everything is a "Point"

- Word embedding
- Phrase embedding
- Sentence embedding
- Word embedding in context of sentence
- Etc

Points are good → reduce sparsity by wt sharing

a single (complex) model can handle all pts

Universal Representations

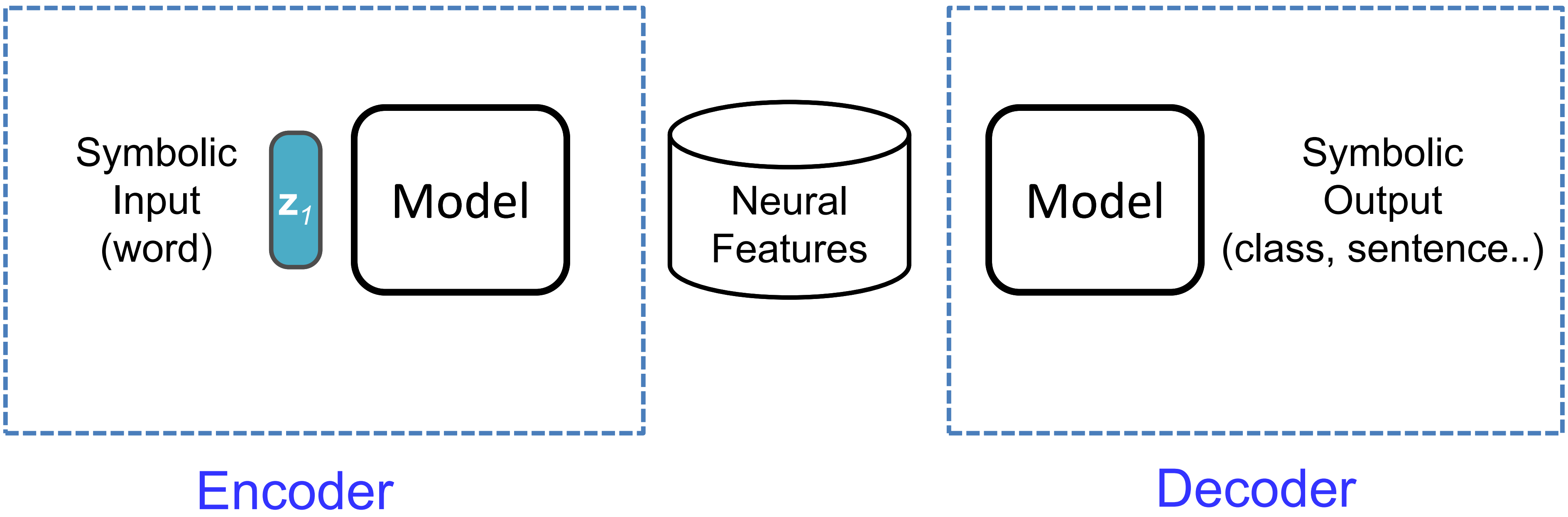
- Non-linearities
  - Allow complex functions
- Put anything computable in the loss function
  - Any additional insight about data/external knowledge

Make symbolic operations continuous

- Symbolic → continuous
  - Yes/No → (number between 0 and 1)
  - Good/bad → (number between -1 and 1)

  - Either remember or forget → partially remember
  - Select from n things → weighted avg over n things

Encoder-Decoder

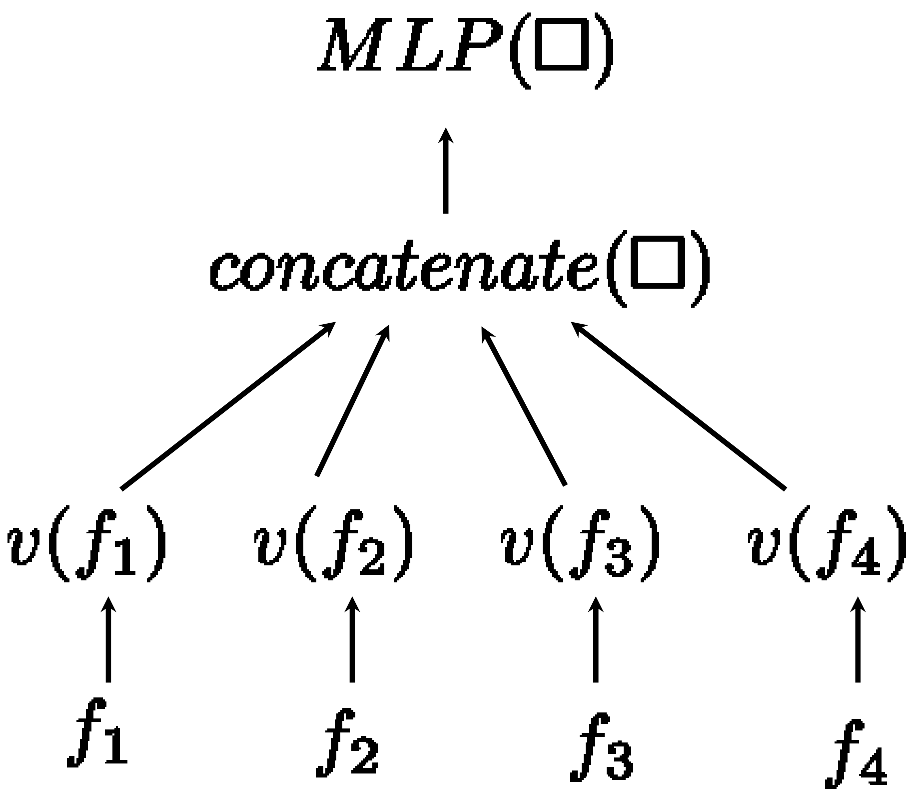


Different assumptions on data create different architectures

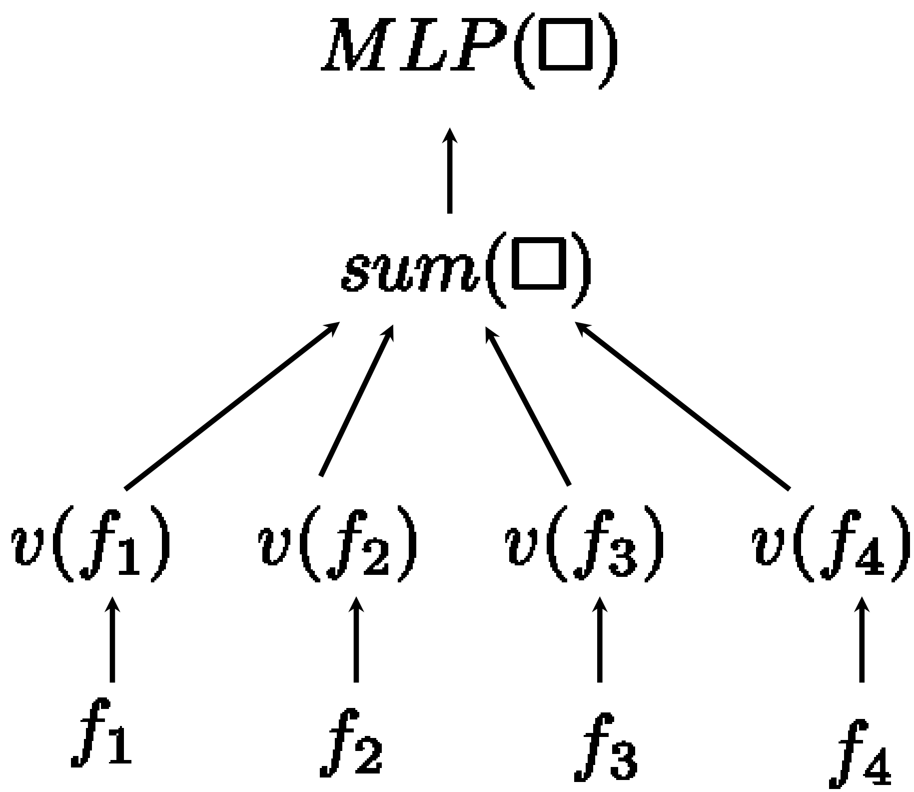
Building Blocks

+	;	.
Matrix-mult	gate	non-linearity

x;y



x+y



Concat vs. Sum

- **Concatenating** feature vectors: the "roles" of each vector is retained.

$$concat\,(v(\text{"}the\text{"}), v(\text{"}thirsty\text{"}), v(\text{"}dog\text{"}))$$

prev word — current word — next word

- Different features can have vectors of different dim.
- Fixed number of features in each example (need to feed into a fixed dim layer).

Concat vs. Sum

- **Summing** feature vectors: "bag of features"

$$sum\,(v(\text{"}the\text{"}), v(\text{"}thirsty\text{"}), v(\text{"}dog\text{"}))$$

word word word

- Different feature vectors should have same dim.
- **Can encode a bag of arbitrary number of features.**

x.y

- degree of closeness
- alignment

- Uses
  - question aligns with answer //QA
  - sentence aligns with sentence //paraphrase
  - word aligns with (~important for) sentence //attention

g(Ax+b)

- 1-layer MLP
- Take x
  - project it into a different space //relevant to task
  - add some scalar bias (only increases/decreases it)
  - convert into a required output

- 2-layer MLP
  - Common way to convert input to output