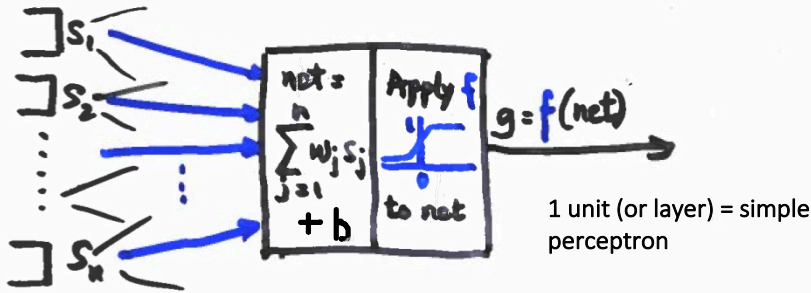


Neural-Network-Based Reasoning

(See the printed notes with that title)

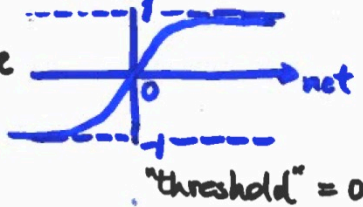
An NN "unit"



e.g., $f(\text{net}) = \frac{1}{1 + e^{-\text{net}}}$ $S_0, \text{threshold} = 0, f(0) = \frac{1}{2}$
 $f(\text{net}) = f(\text{net})(1 - f(\text{net}))$

e.g., $f(\text{net}) = \tanh(\text{net})$

(if we allow negative signals)

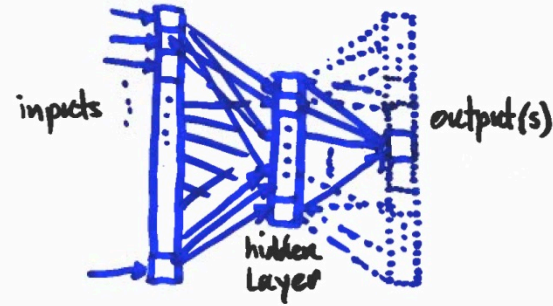


$$\frac{1 - e^{-2x}}{1 + e^{-2x}}$$

der: $\frac{4}{(e^x + e^{-x})^2}$
 $= \frac{4}{(1 + \tanh(x))^2}$

Note the alternation of a linear operation and a nonlinear one.

Typical 3-layer architecture: (feed-forward)



for classification, a single output unit suffices.

For pattern transformation, there may be many outputs.

Supervised Learning

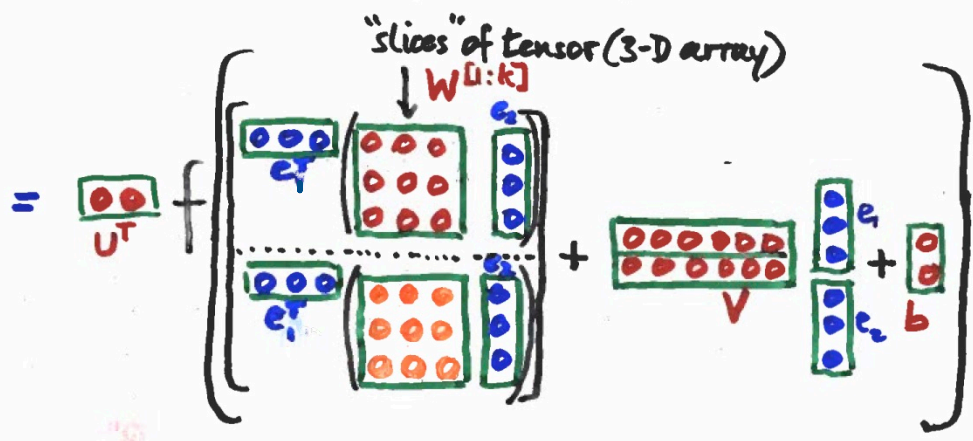
Adjust weights to minimize squared error over the (labeled) outputs of the training corpus.

Make use of the partial derivatives of the above total error w.r.t. each weight. Can use global gradient methods or backpropagation.

It turns out, the deeper (more layers) the better. Often convolutional layers are used, and other special layers connected in a specific way to the previous layer, and all units use the same weights.

Socher et al. - Reasoning with tensor networks

$$g(e_1, R, e_2) = u^T f(e_1^T W^{[1:k]} e_2 + V[e_1] + b)$$



$|e_1| = |e_2| = d$ (here, 3, in practice 100)
 depth of weight array $W^{[1:k]} = k$ (here 2, in practice 4)

e.g., relation $R = \text{nationality}$, instance
 (Pablo Picasso, nationality, Spain)

e_1 co-occurrence vector "embeddings" e_2 co-occurrence vector

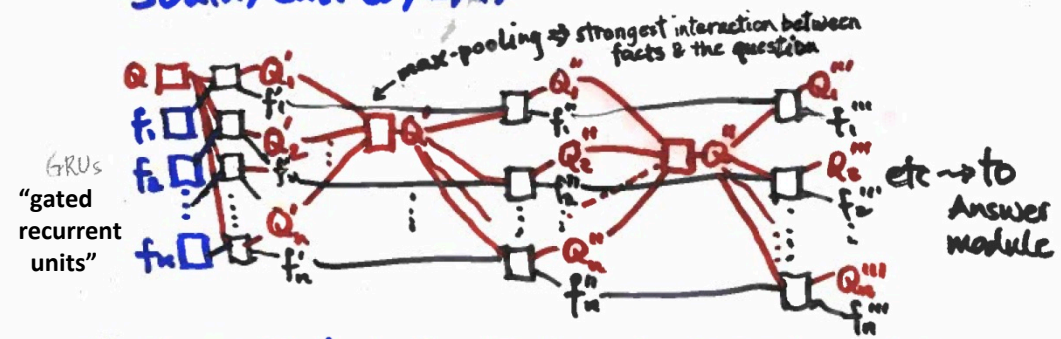
Can infer this from Freebase Triples (has nearly 2×10^9)
 About 90% correct on held-out test (non) examples.
 Basically, this is **similarity inference (analogy)**

Peng et al., Towards NN-based reasoning*

1. The office is east of the hallway
2. The kitchen is north of the office
3. The garden is west of the bedroom
4. The office is west of the garden
5. The bathroom is north of the garden

K B a
H O G B d

How do you go from the kitchen to the garden?
 South, east (by 2, 4)



Guillaume-Bert et al., First-order Logic Learning**

- (A) From examples and non-examples of $P(x, f(x)) \wedge Q(g(x)) \Rightarrow R(h(f(x)))$ (numerically encoded)
 (where pos^{ve} examples have $x \in \{a, b, c, \dots\}$) learn this rule.
- (B) White king & white rook vs black king on chess board; is any piece under attack? (logically coded)
- (C) 10 toy trains with up to 4 cars, each carrying up to 3 objects. 5 are going east & 5 are going west. Learn from 9 trains at a time, test on the 10th. Full success
 Conventional NN units & arithmetic units are used.

* <https://arxiv.org/pdf/1508.05508.pdf>

** https://openaccess.city.ac.uk/id/eprint/297/2/First-order_Logic_Learning_in_Artificial_Neural_Networks.pdf