Deep Learning

with No maTh

Philipp Gabler

2022-06-28

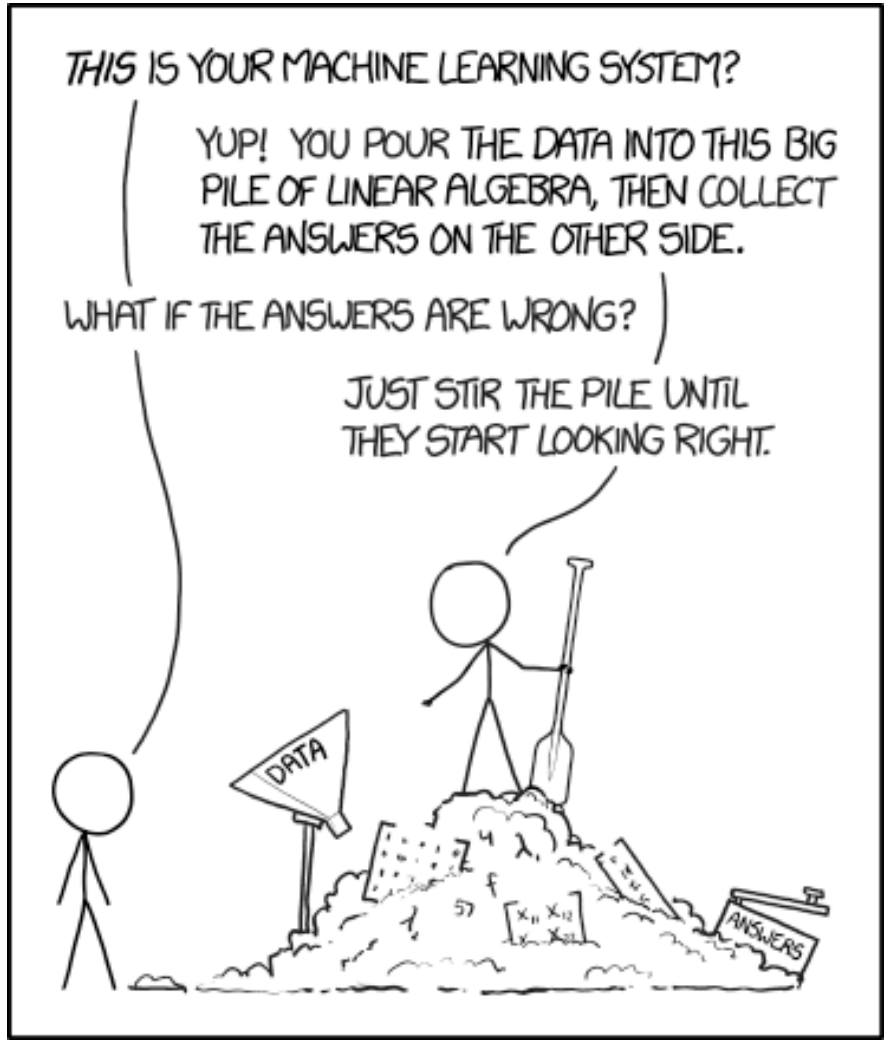
DEEP LEARNING

WITH ~~NO~~ MATH

AS LITTLE AS POSSIBLE

PHILIPP GABLER

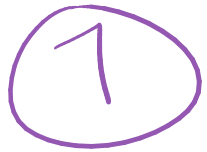
2022-06-28



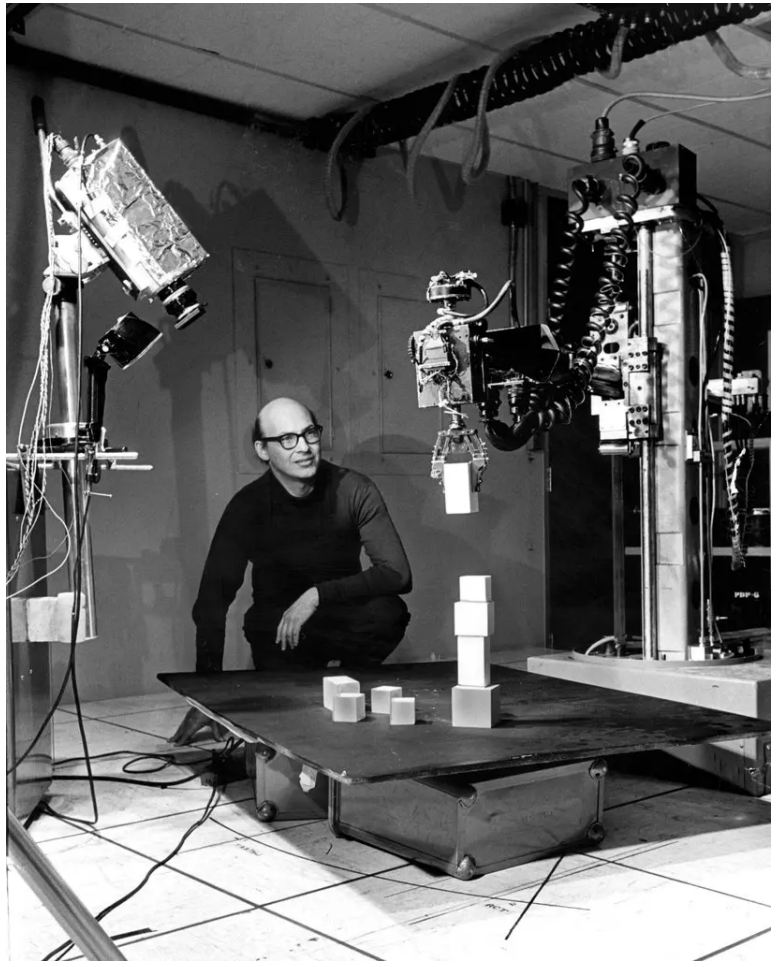
https://xkcd.com/1838/

OUTLINE

1. A BIT OF HISTORY
2. WHAT PROBLEM(S) DOES DL SOLVE? — COMPUTATIONAL
3. WHAT DO SOLUTIONS LOOK LIKE? — ALGORITHMIC
4. HOW ARE SOLUTIONS IMPLEMENTED? — IMPLEMENTATIONAL
5. SOME LANGUAGE-RELATED USE CASES



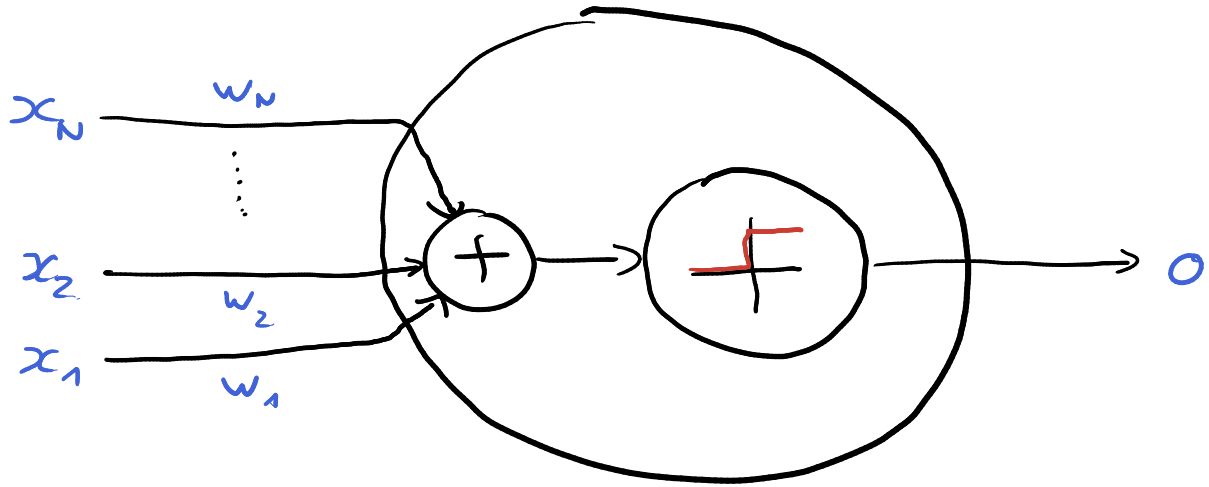
A BIT OF HISTORY



https://www.nytimes.com/2016/01/26/business/marvin-minsky-pioneer-in-artificial-intelligence-dies-at-88.html

Ἐν ἀρχῇ ἦν τὸ PERCEPTRON

- McCULLOCH & PITTS, 1948: NEURON MODEL



- ROSENBLATT, 1958: ENTHUSIASM

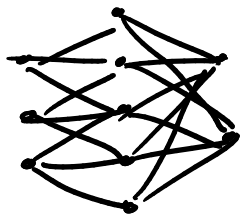
- MINSKY & PAPERT, 1969: SHOCK OF "PERCEPTRONS"

❄ AI WINTER ❄

PROBLEM OF LINEAR SEPARABILITY

NEW SPRING

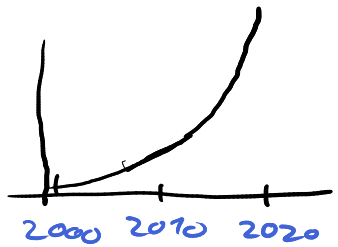
- RUMELHART & MCCLELLAND (1986): PDP
  - ↳ CONNECTIONISM
  - ↳ BACKPROPAGATION
  - ↳ SINGLE-LAYER → MULTI-LAYER

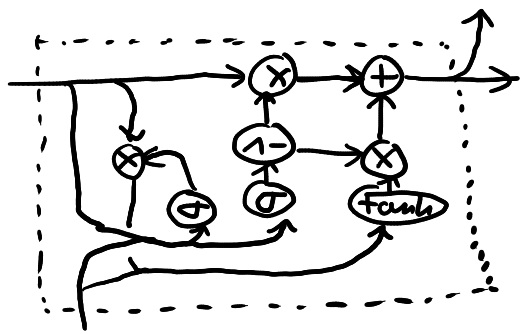


- STATISTICAL LEARNING / MACHINE LEARNING OVER (SYMBOLIC) AI

TOWARDS DEEP LEARNING

- MORE LAYERS
- MORE DATA
- MORE COMPUTE (MOORE'S LAW)
- BETTER TUNING & TRAINING



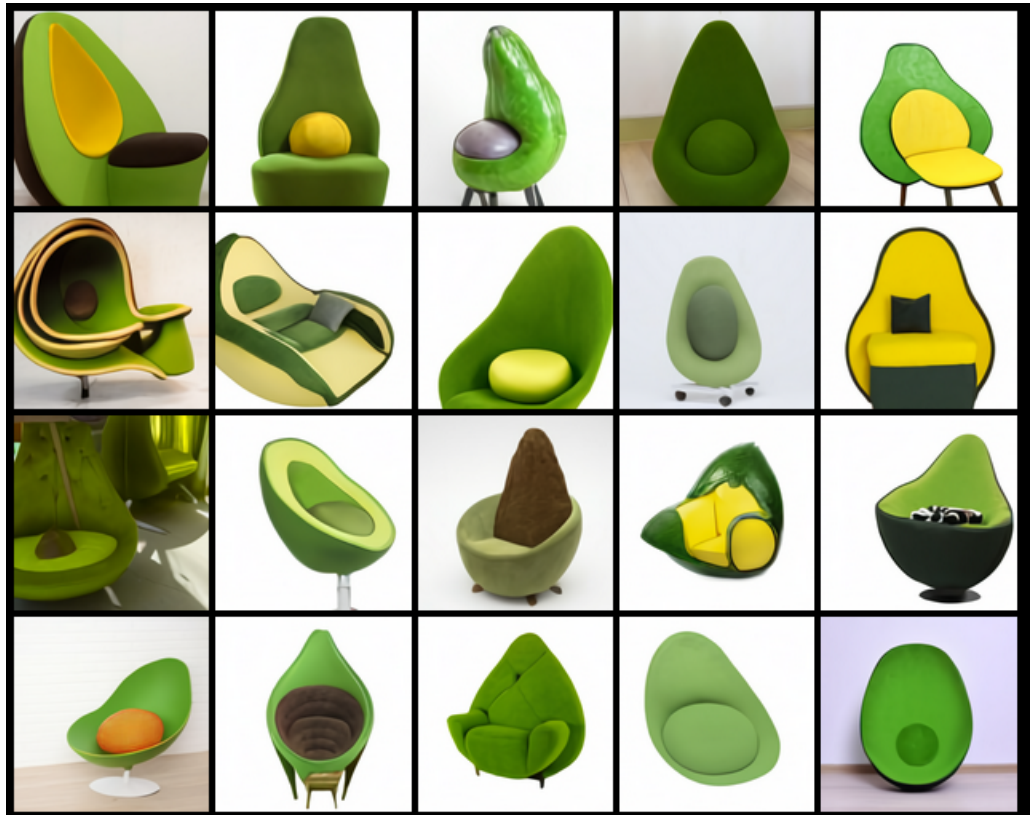


TAKE-AWAYS 1

- ARTIFICIAL – NO NEURONS ANYMORE!
- AI AND ML HAVE SPLIT – DL ≠ INTELLIGENCE
- GROWTH OF TECHNOLOGY IMPORTANT
- DL IS SUB SYMBOLIC (≃ CONNECTIONISM)

WHAT PROBLEMS DOES DEEP LEARNING SOLVE?

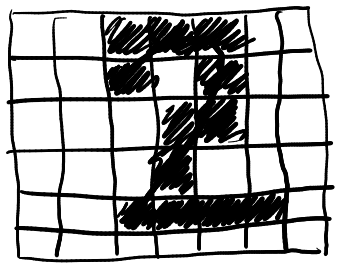
"AN ARMCHAIR IN THE SHAPE OF AN AVOCADO"



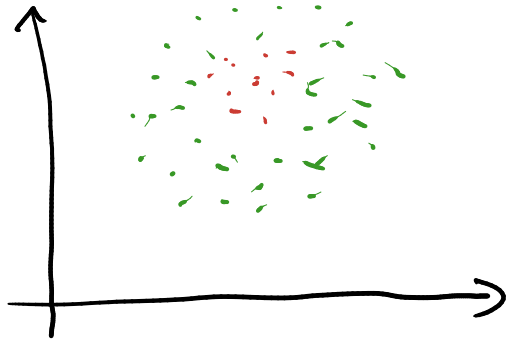
https://openai.com/blog/dall-e/

CHARACTERISTICS OF DEEP LEARNING

HIGH-DIMENSIONAL

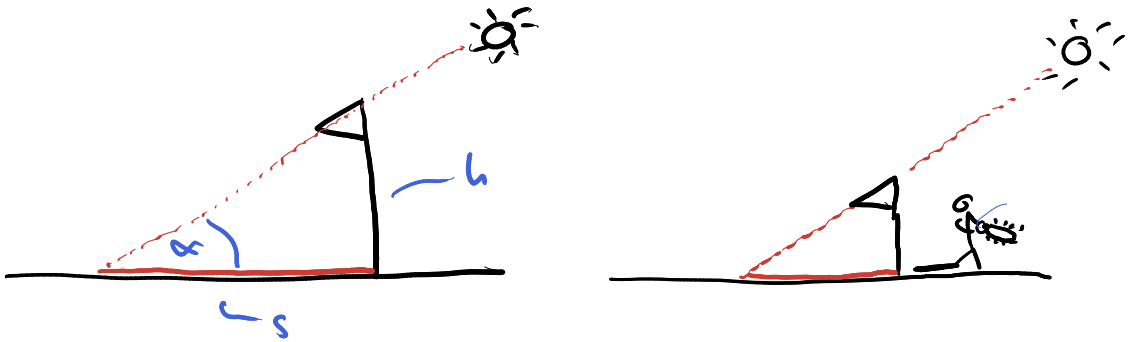


NON-LINEAR

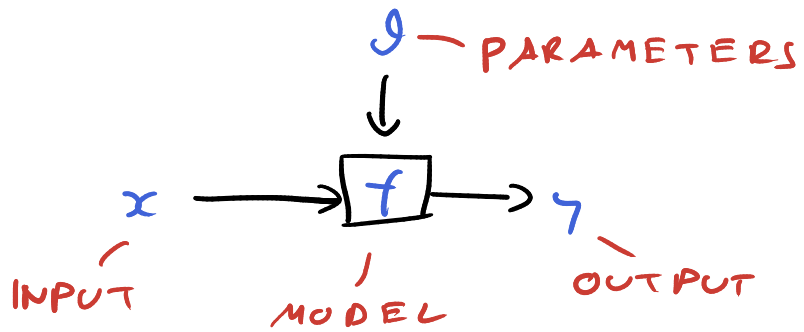


PREDICTION

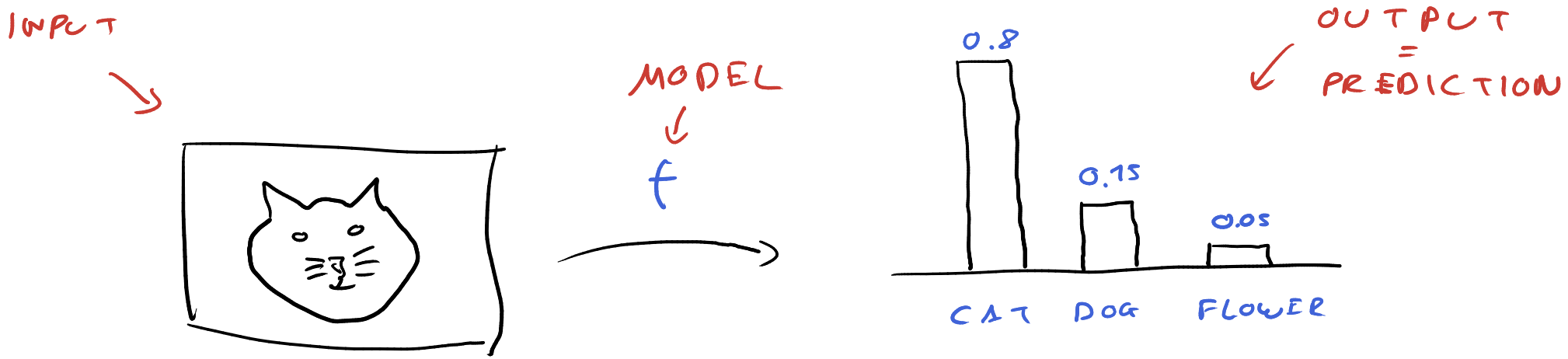
ASSOCIATIVE



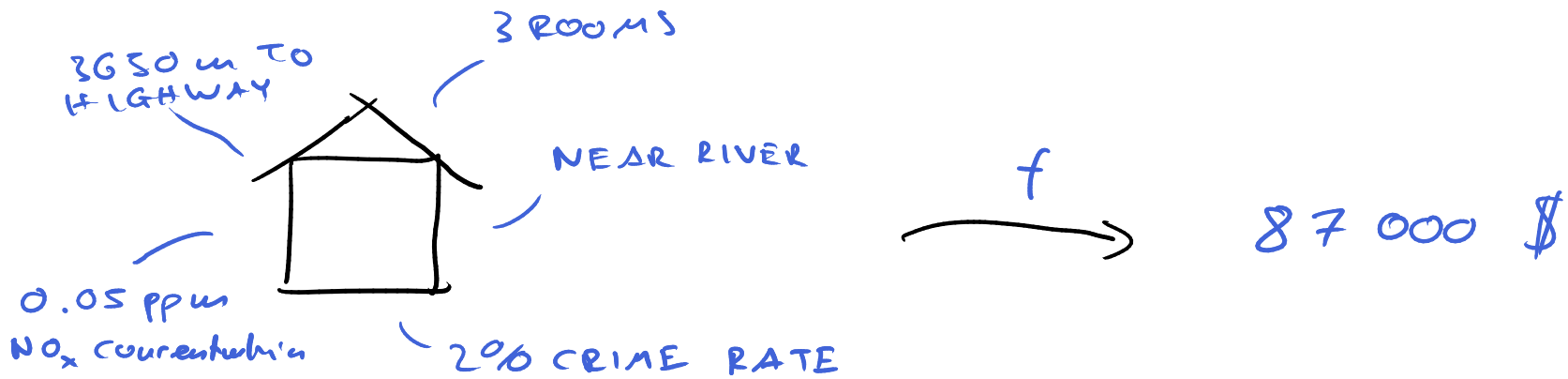
END-TO-END TRAINING, PARAMETRIC



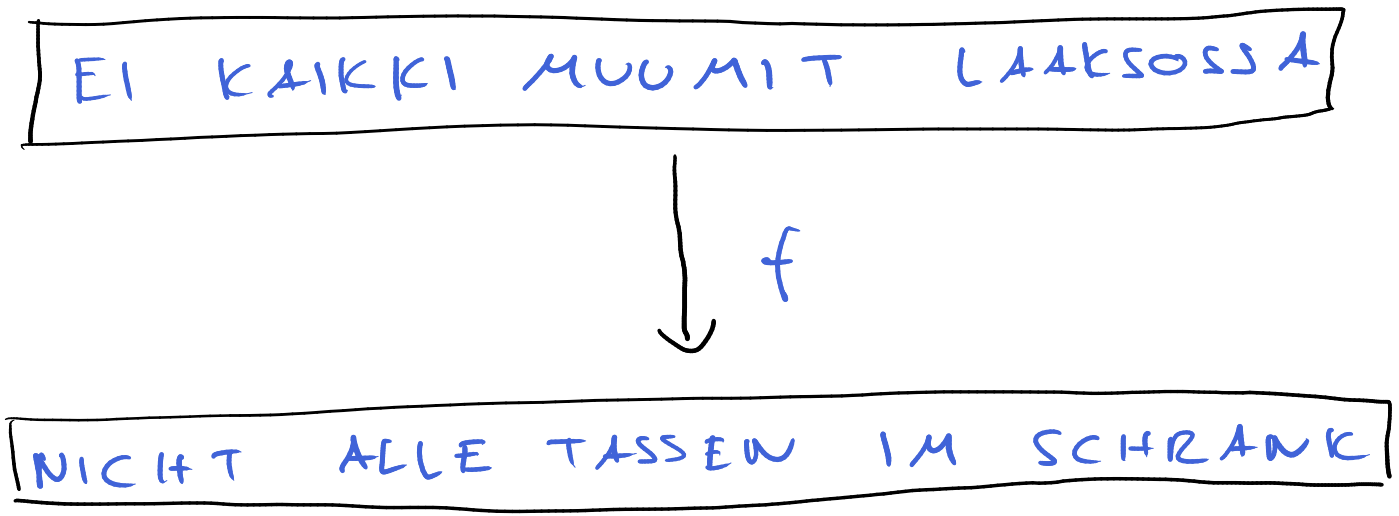
CLASSIFICATION



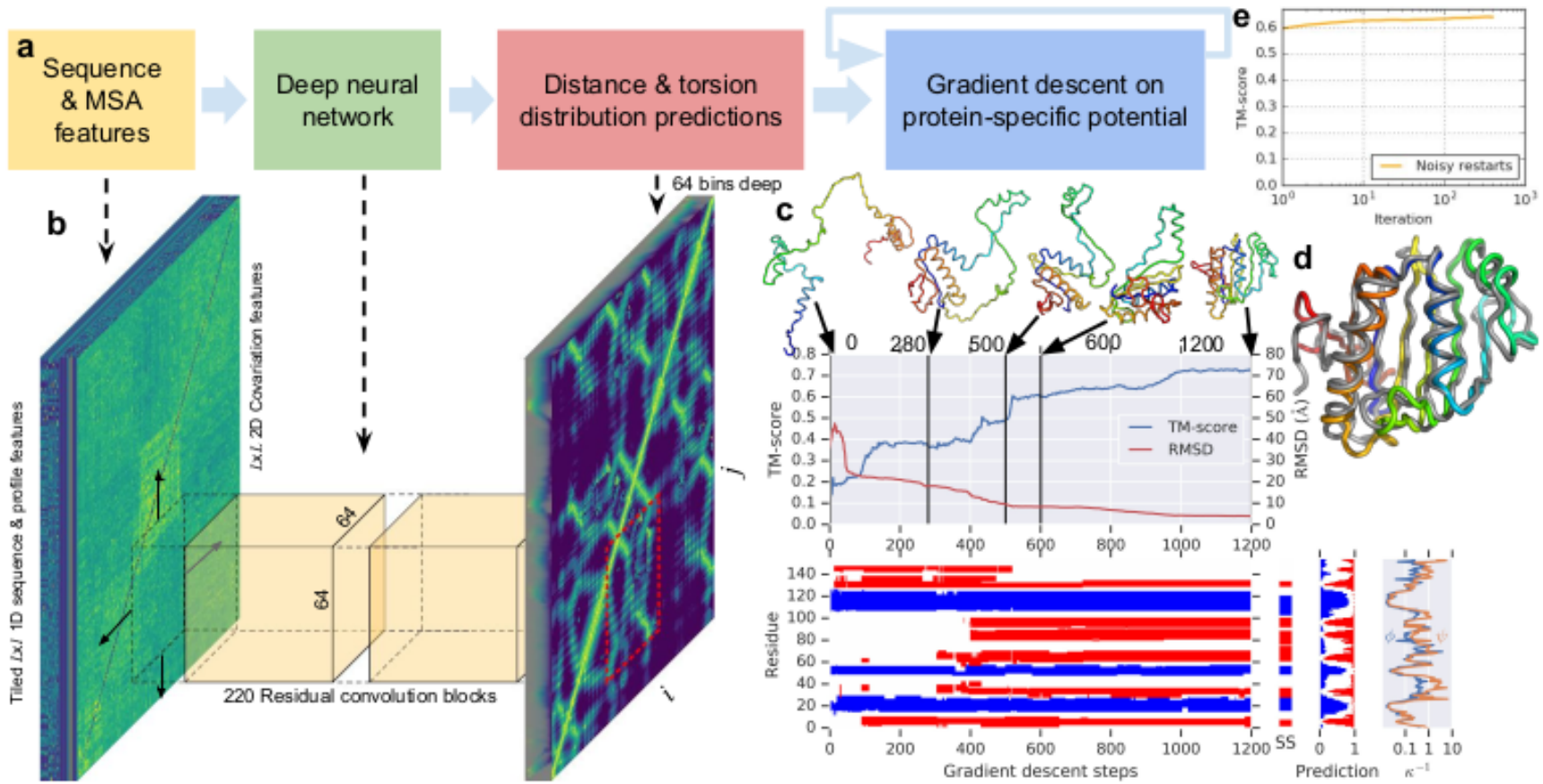
REGRESSION


3650 m TO HIGHWAY
3 ROOMS
NEAR RIVER
0.05 ppm
NO_x concentration
20% CRIME RATE
f
87 000 $

SEQUENCE PREDICTION



HIGH-DIMENSIONAL MODELLING



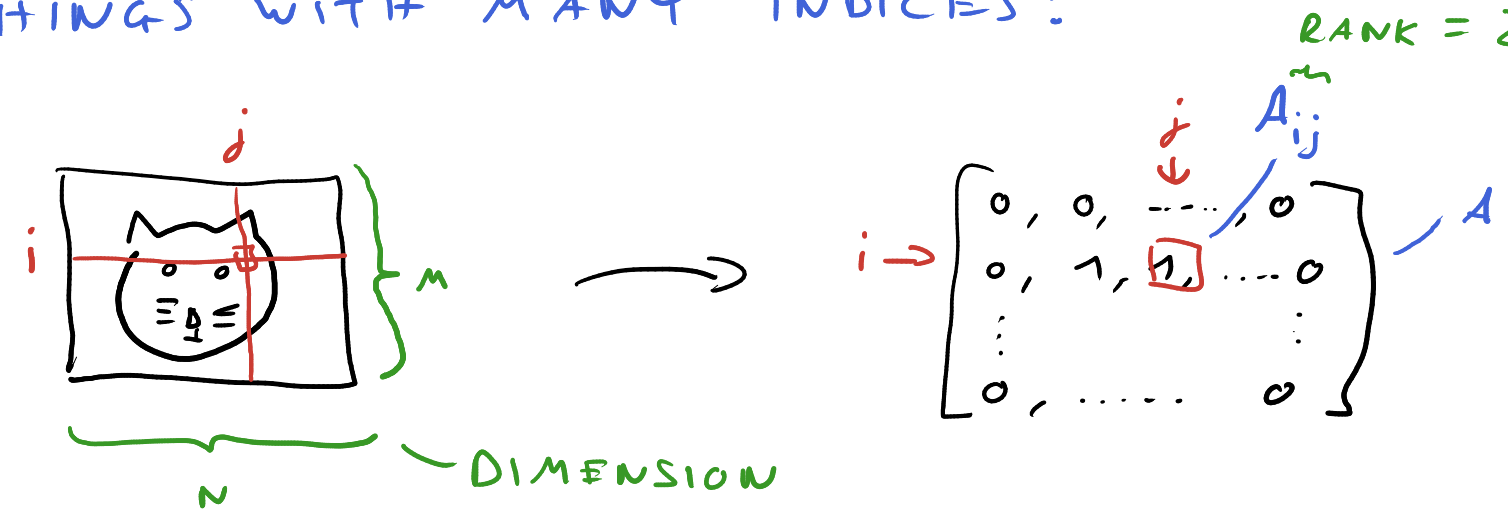
f

https://doi.org/10.1038/s41586-019-1923-7

TENSORS

TensorFlow

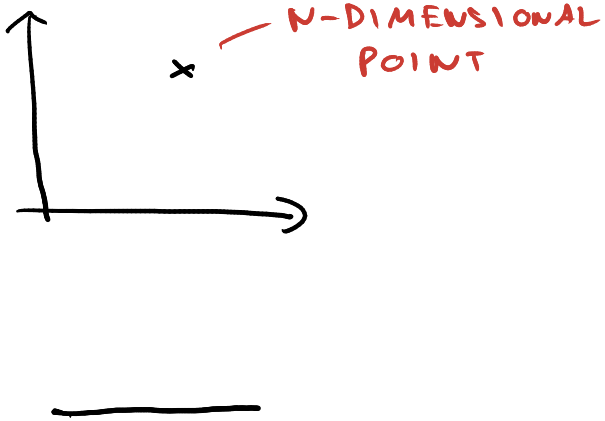
JUST THINGS WITH MANY INDICES:



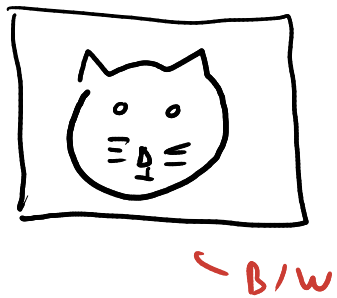
EVERYTHING IS LINEAR ALGEBRA!

TENSOR SHAPES

RANK 1

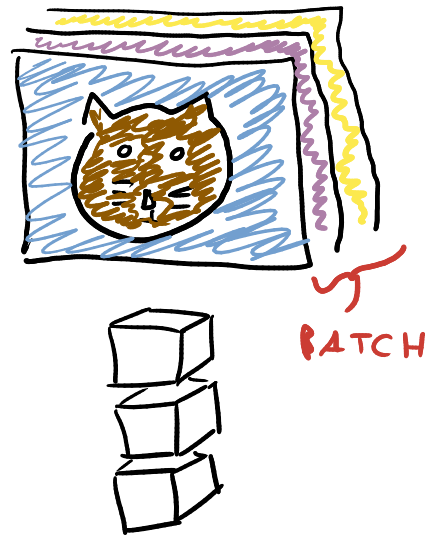


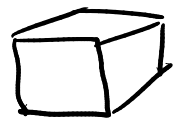
RANK 2



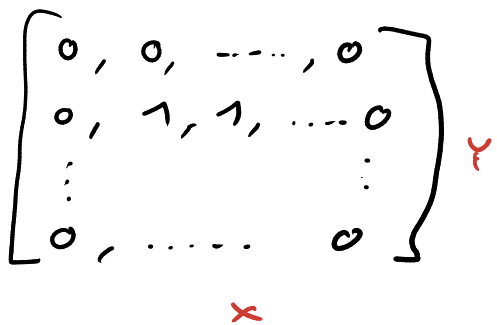
RANK 3

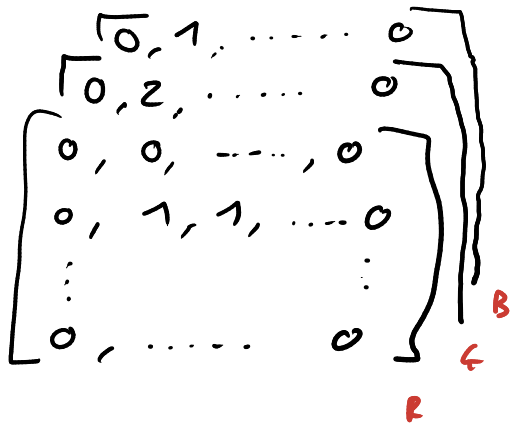
RANK 4



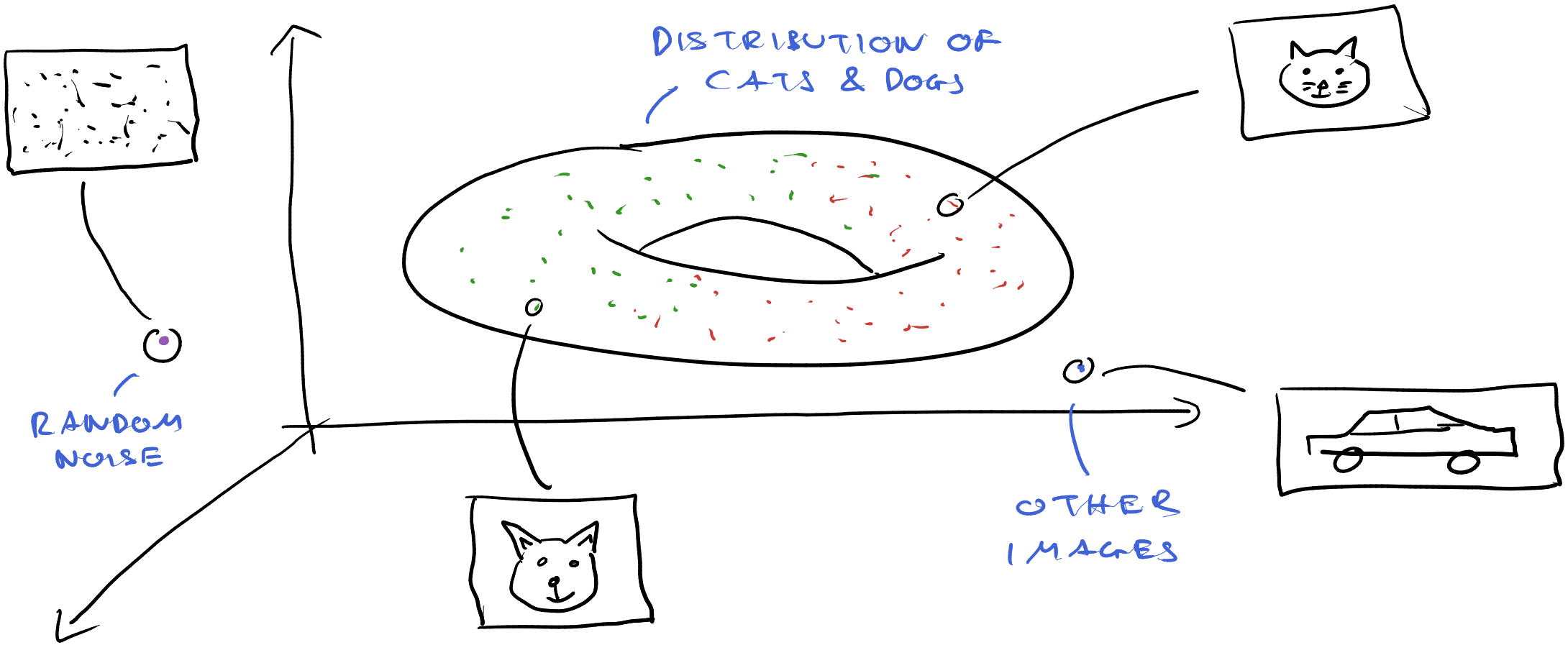


[2.0, -0.3]

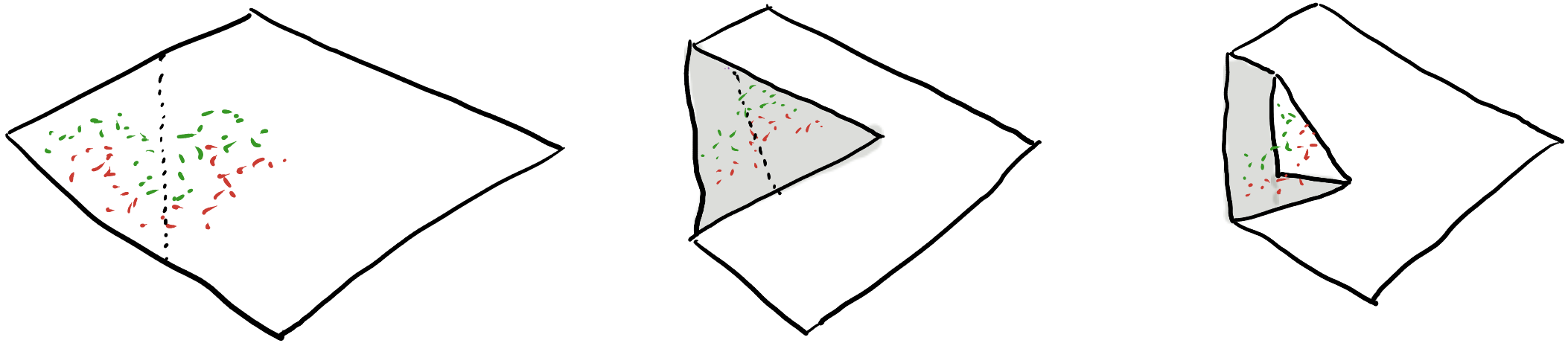




MANIFOLD HYPOTHESIS

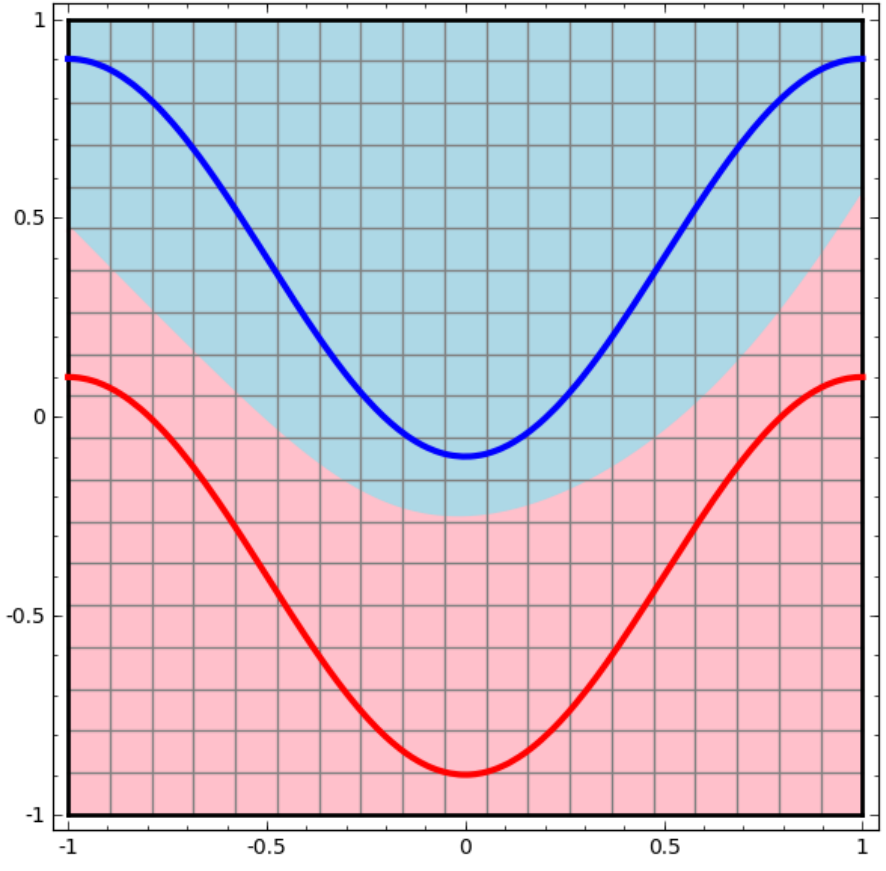


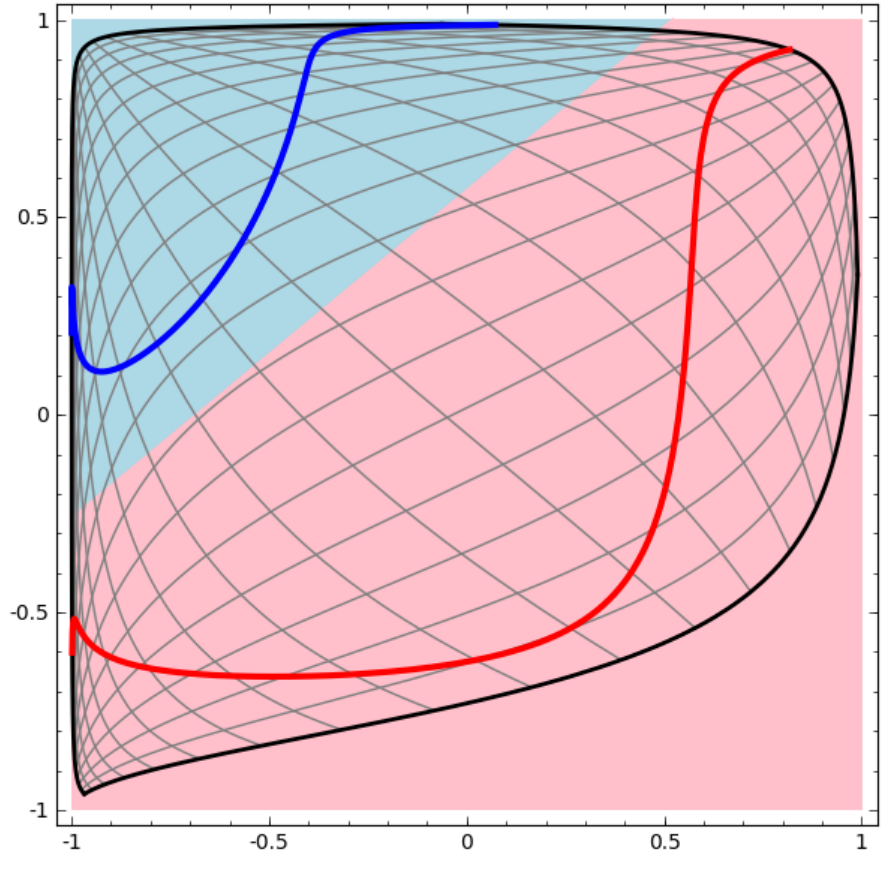
Space Folding



https://doi.org/10.48550/arXiv.1402.1869

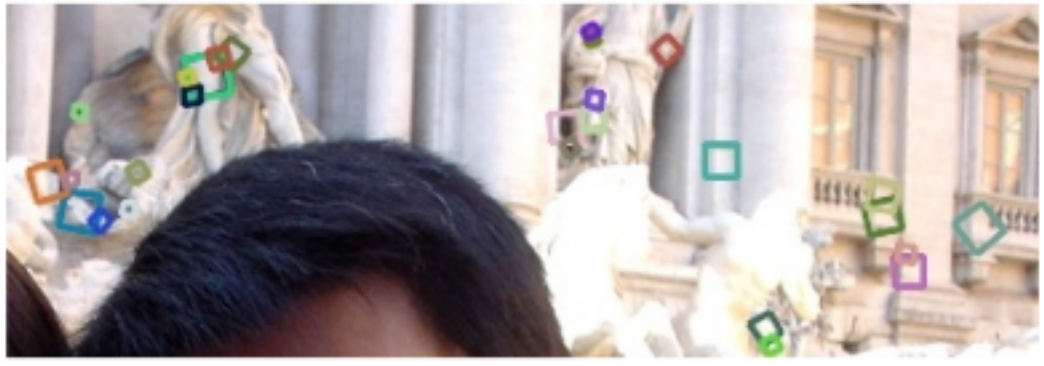
NONLINEAR TRANSFORMATIONS



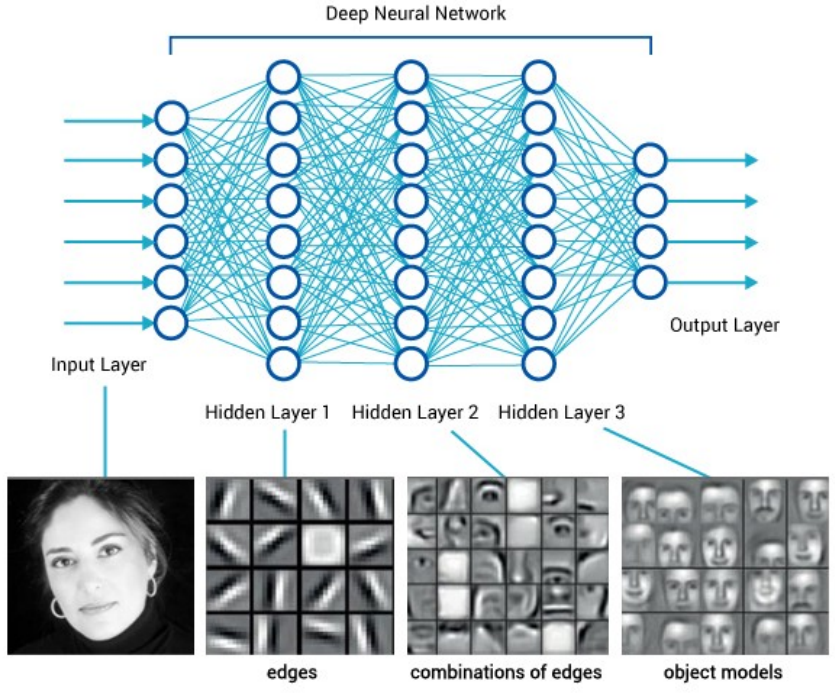


From Features to Layers

Traditional



Deep Learning



https://medium.com/diaryofawannapreneur/deep-learning-for-computer-vision-for-the-average-person-861661d8aa61

WHAT DO SOLUTIONS LOOK LIKE?

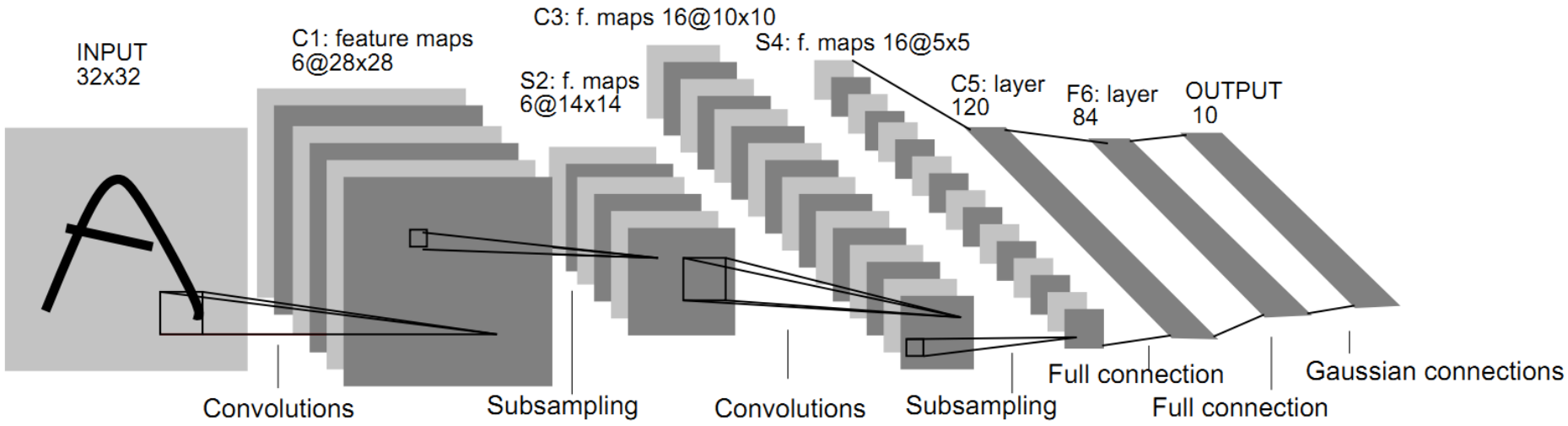
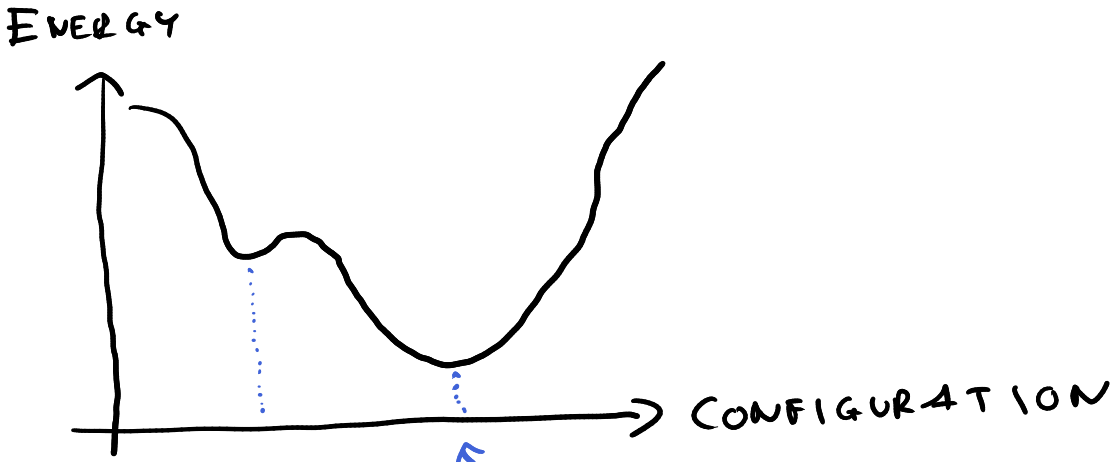


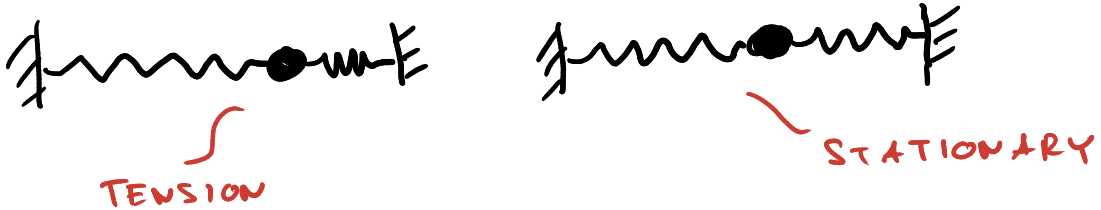
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

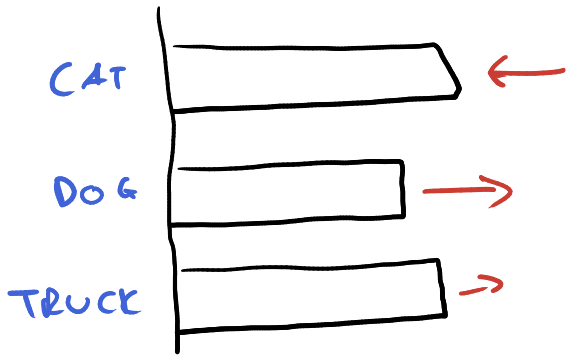
https://doi.org/10.1109/5.726791

ENERGY MINIMIZATION

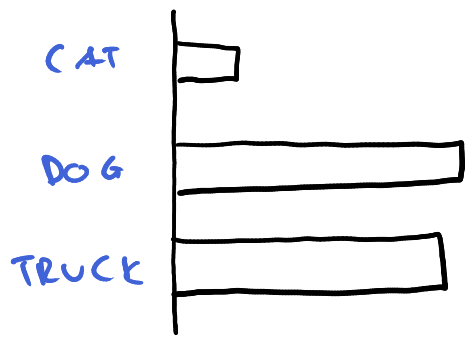


$E(\text{DATA}, \vartheta_0) > E(\text{DATA}, \vartheta_1)$

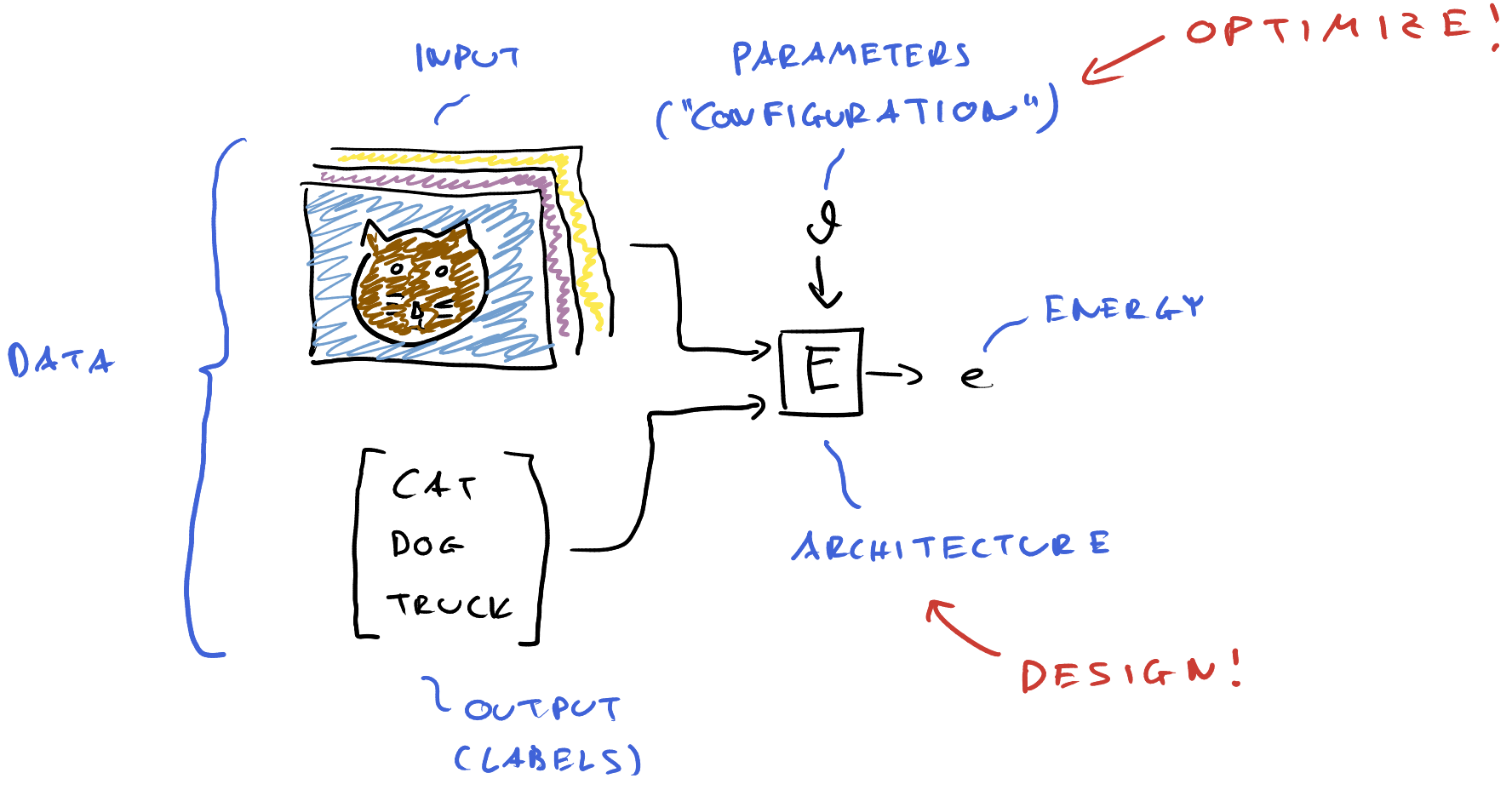




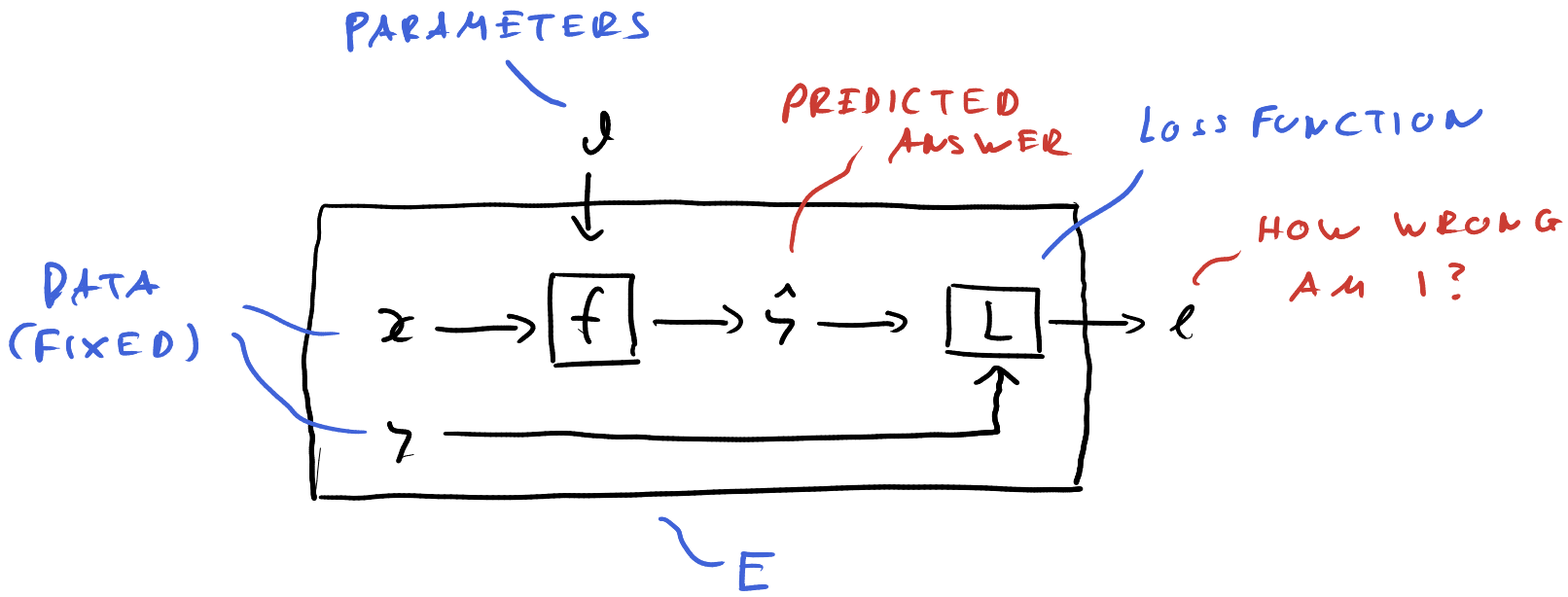
LEARNING



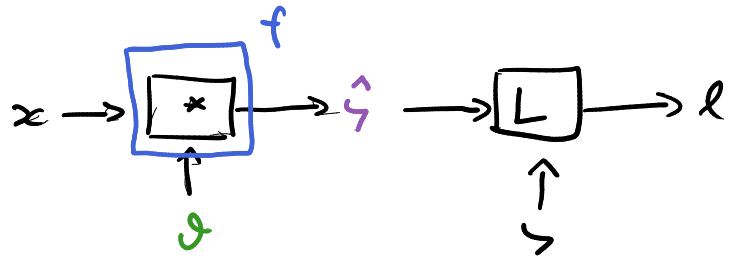
PARAMETRIZED FUNCTIONS

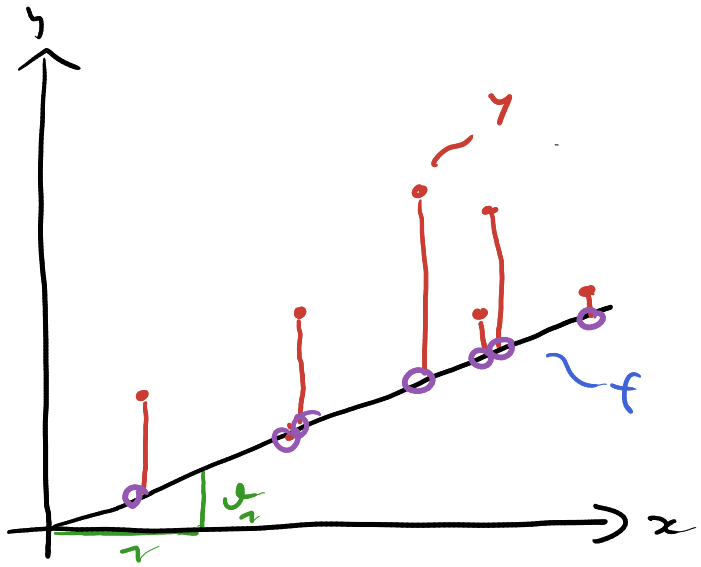


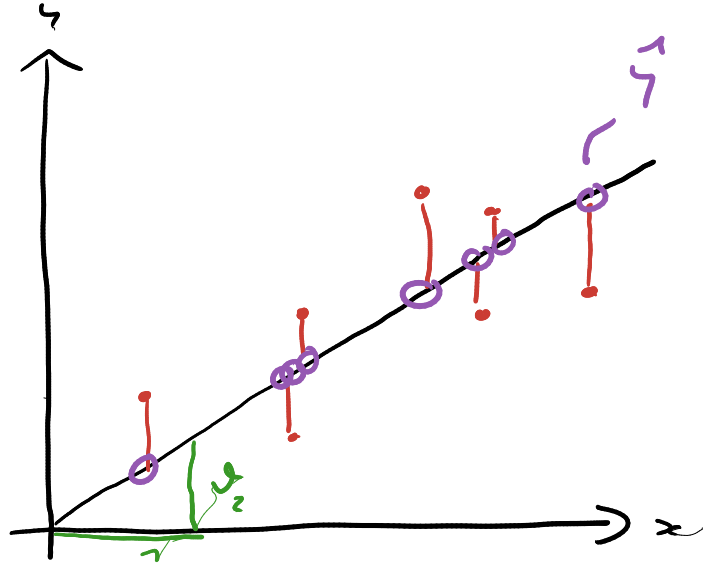
LOSS AS ENERGY



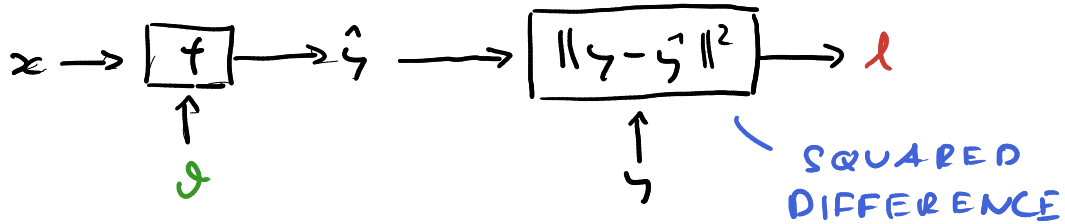
LINEAR REGRESSION

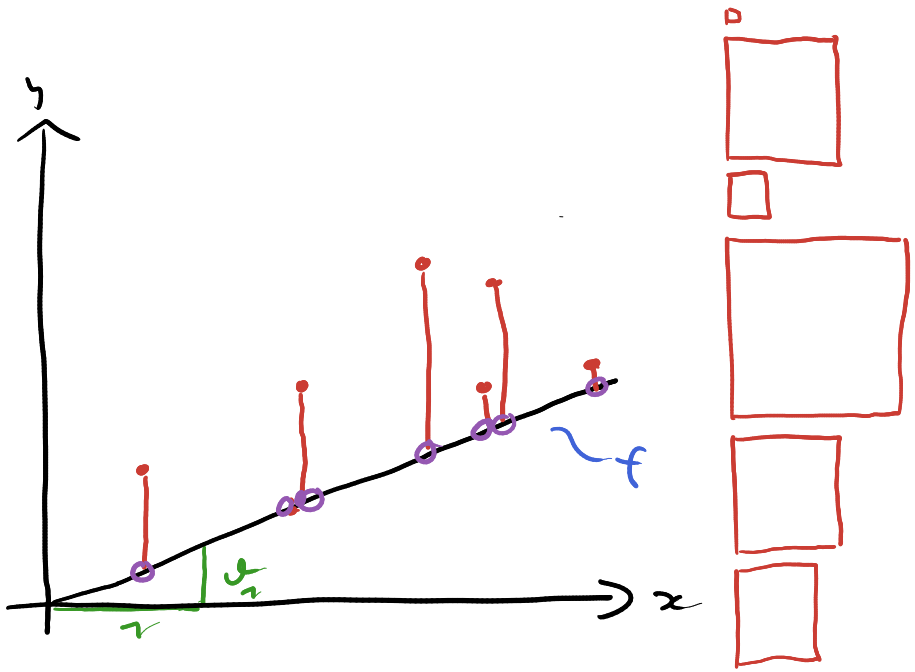


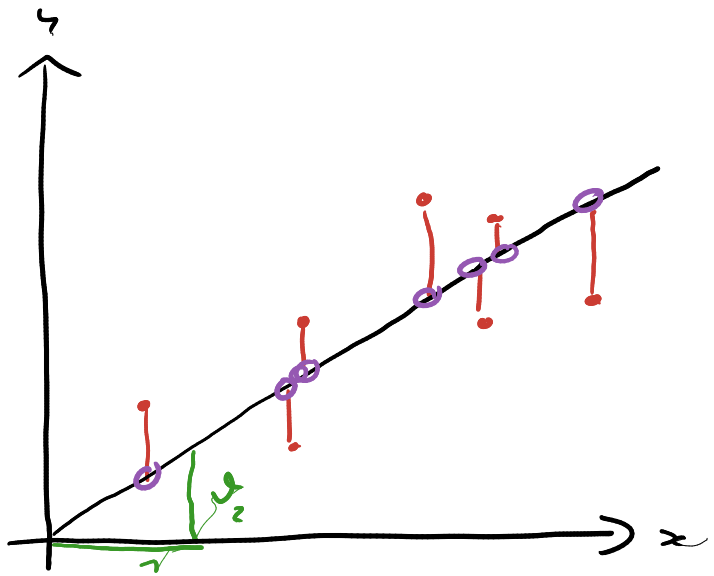




SQUARED DIFFERENCE LOSS

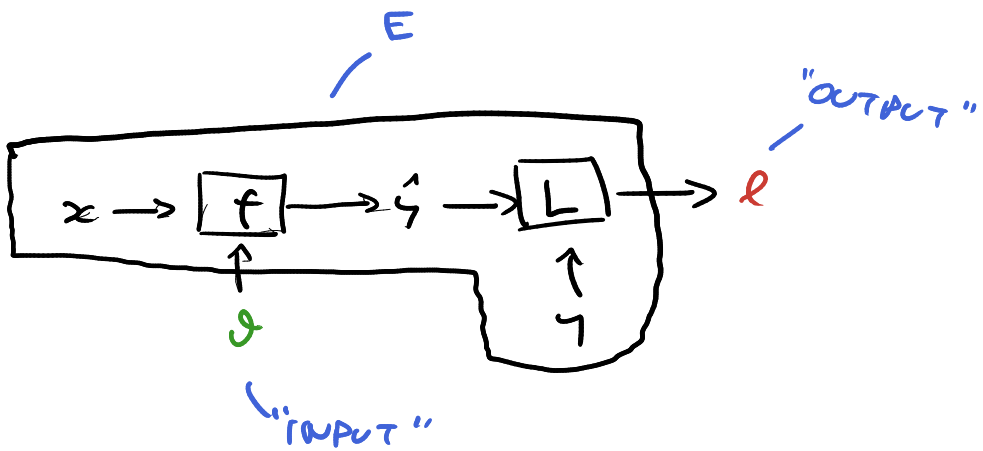


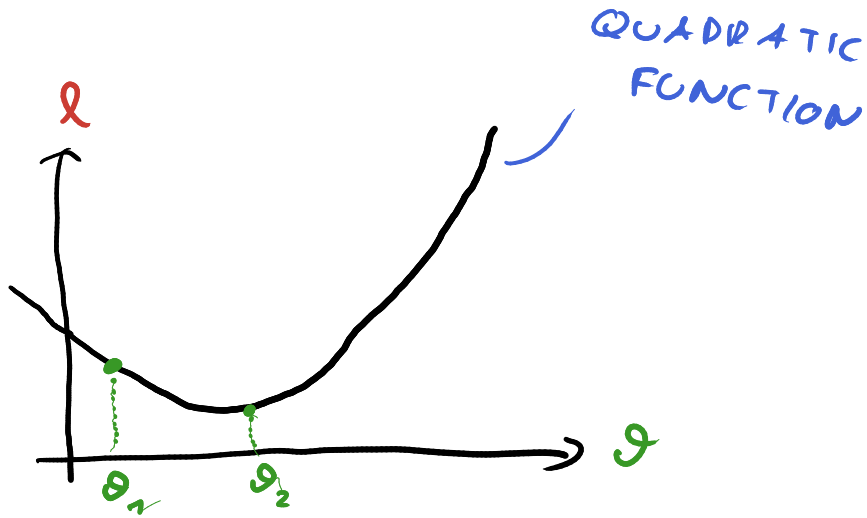




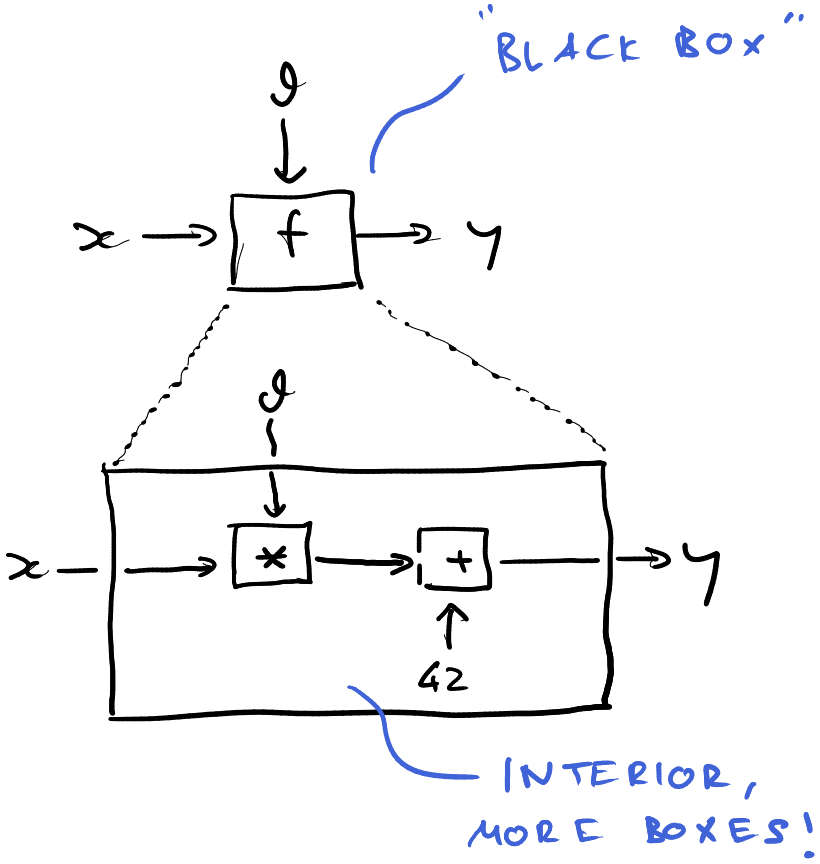
"LEAST SQUARES"...

LEAST SQUARES OPTIMIZATION

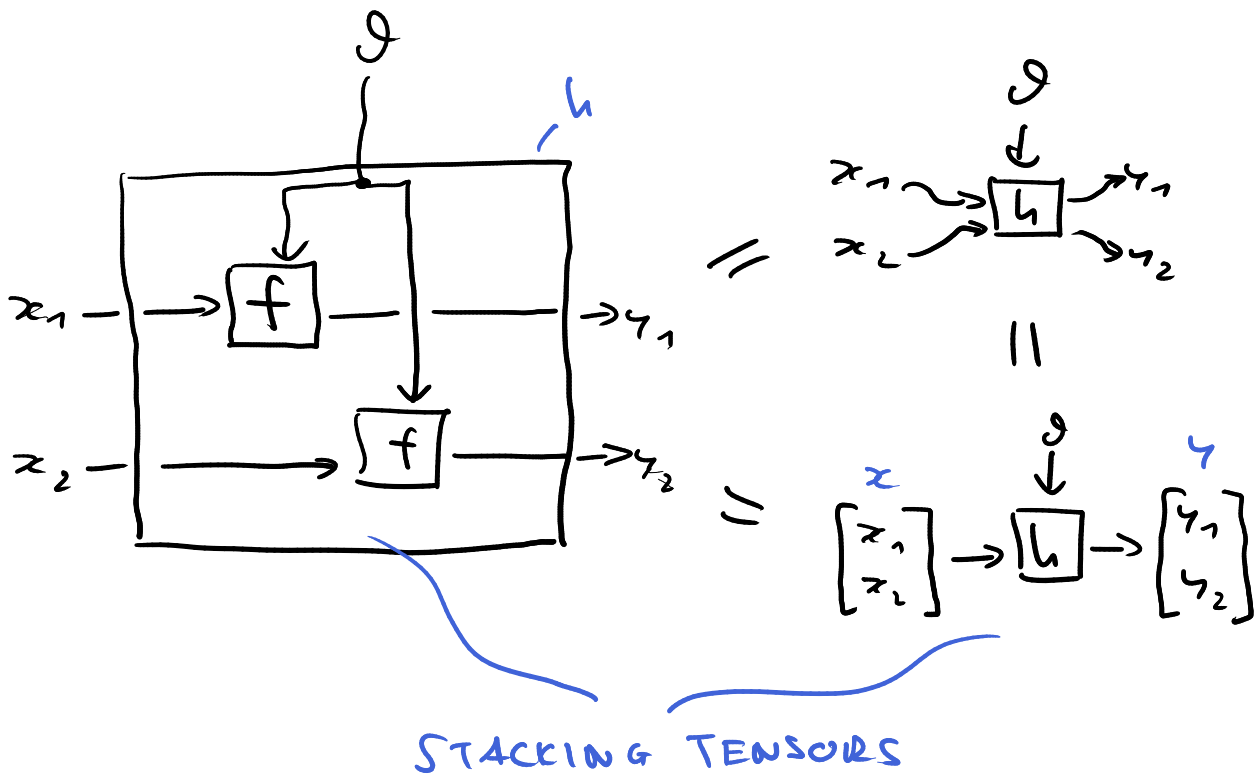




LEGO FOR FUNCTIONS

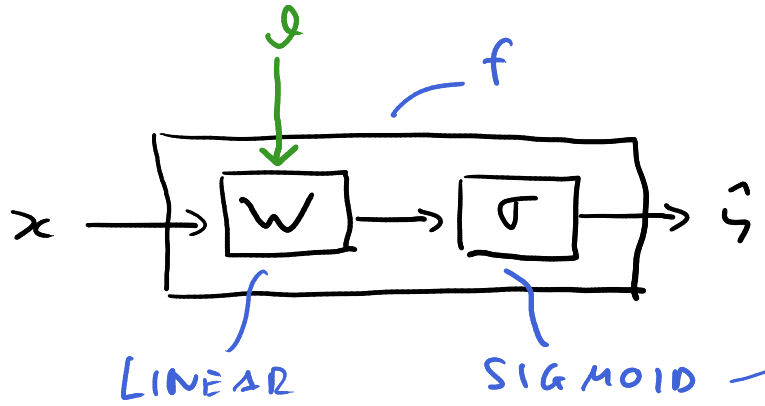


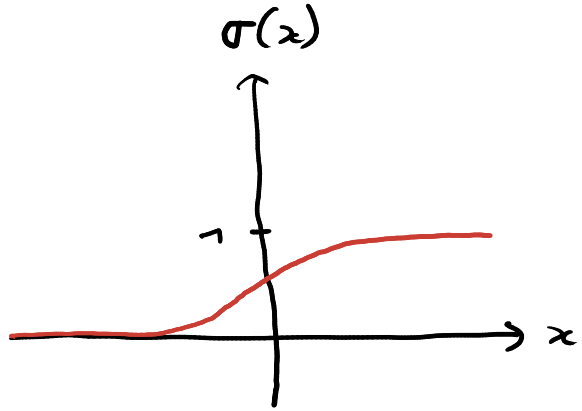
$(x, \theta) \mapsto \theta x + 42$

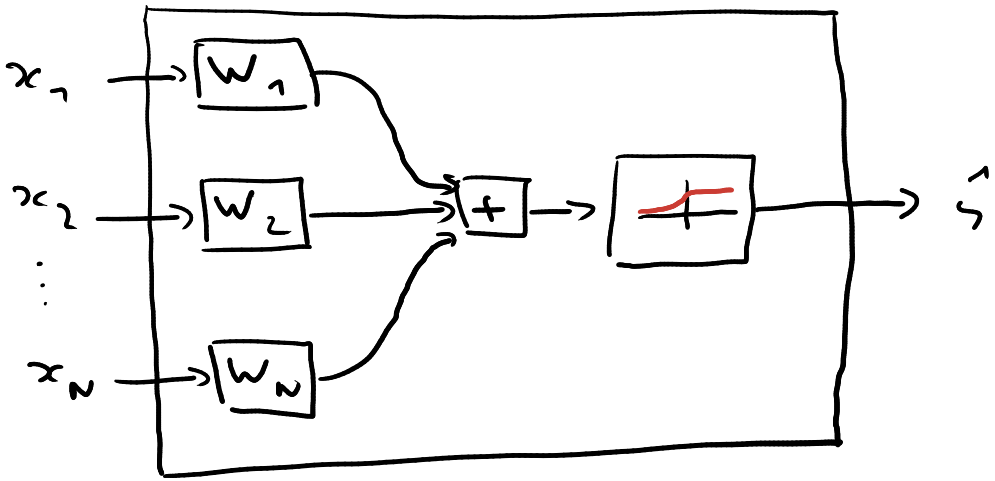


$(x, \theta) \mapsto [f(x_1, \theta), f(x_2, \theta)]$

LOGISTIC REGRESSION

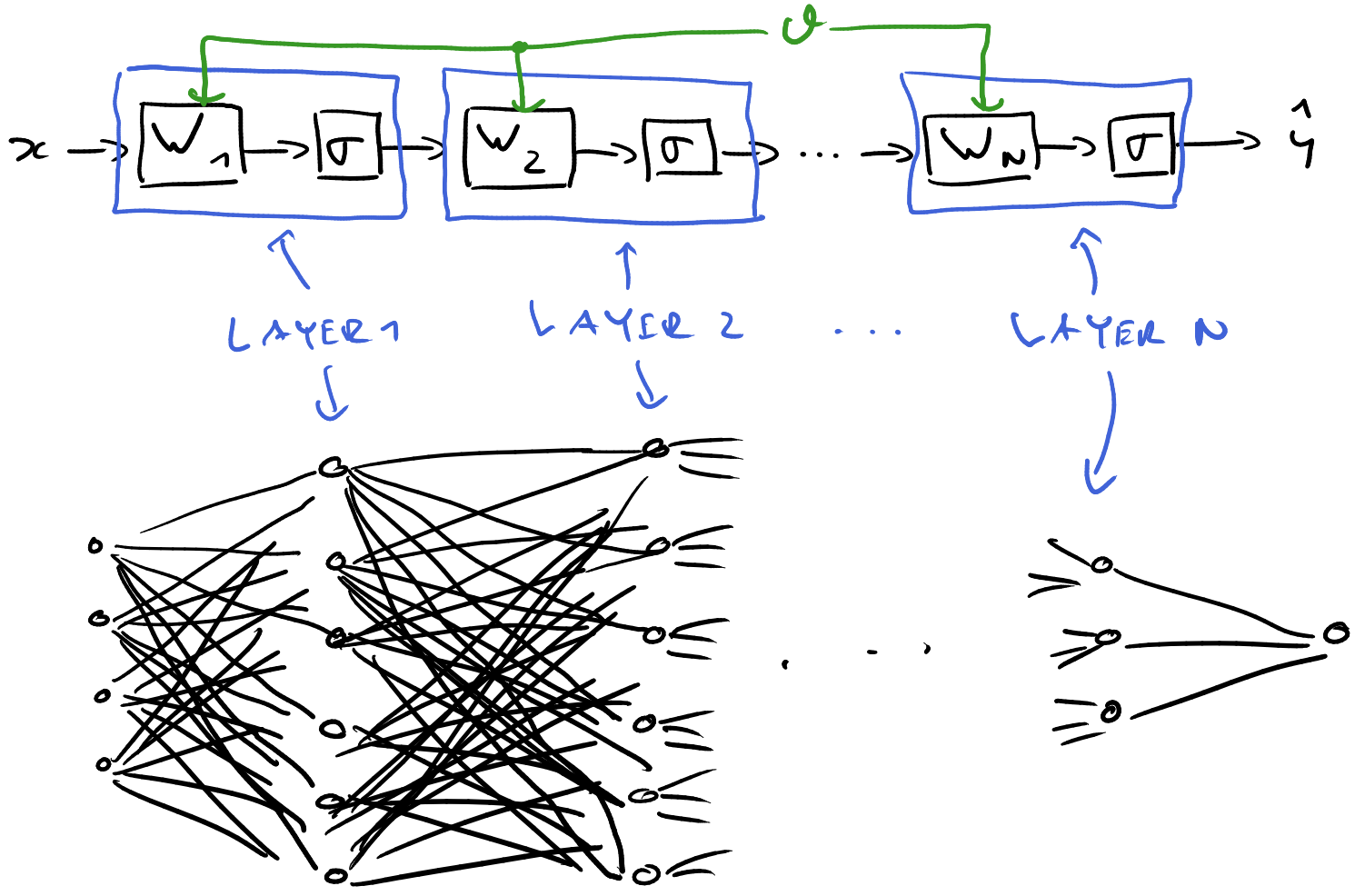






REMEMBER THE PERCEPTRON?

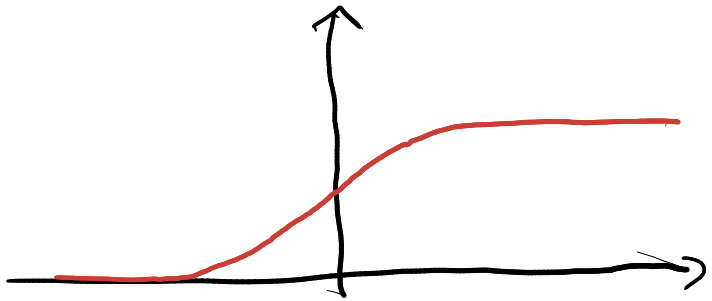
DEEP LEARNING = DEEP F



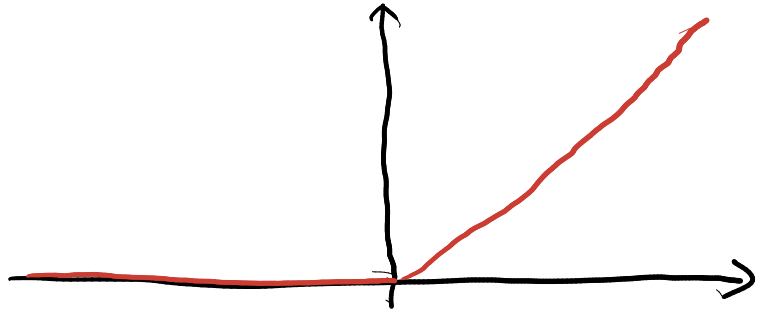
MANY PARAMETERS!

"MULTI-LAYER PERCEPTRON" / FEEDFORWARD NETWORK

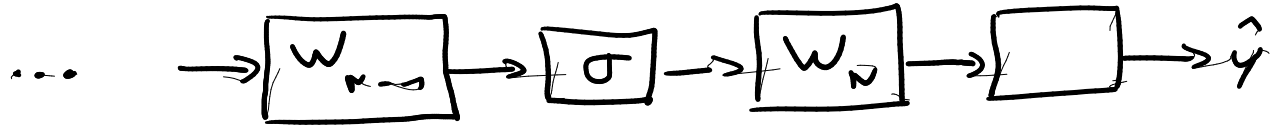
DIFFERENT NONLINEARITIES



SIGMOID

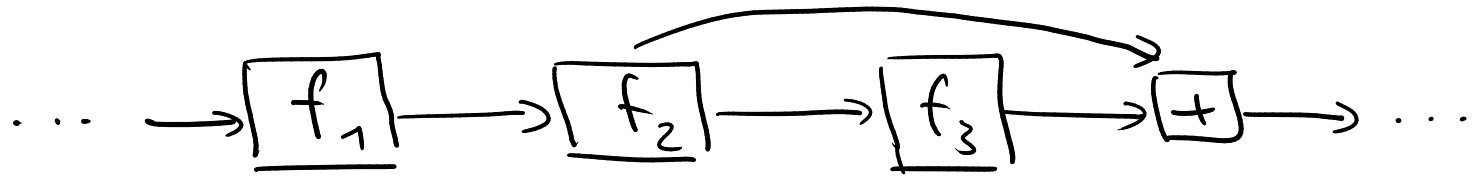


RELU

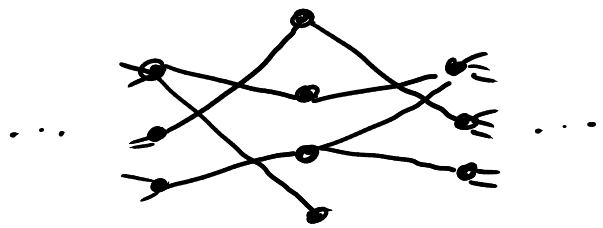


LINEAR OUTPUT

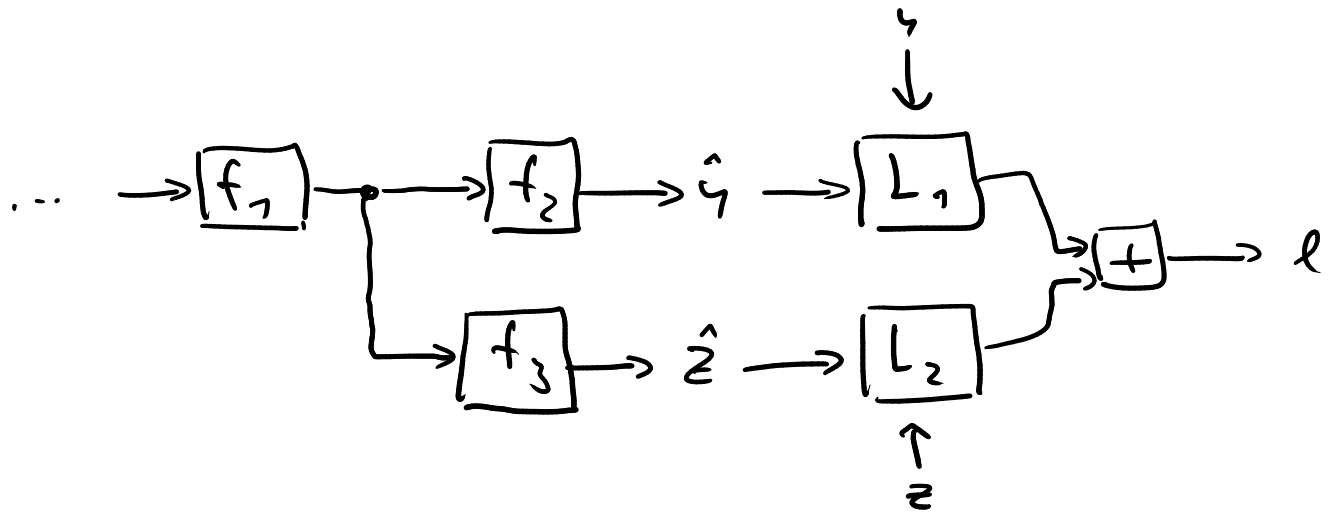
MORE VARIATION



SKIP CONNECTIONS

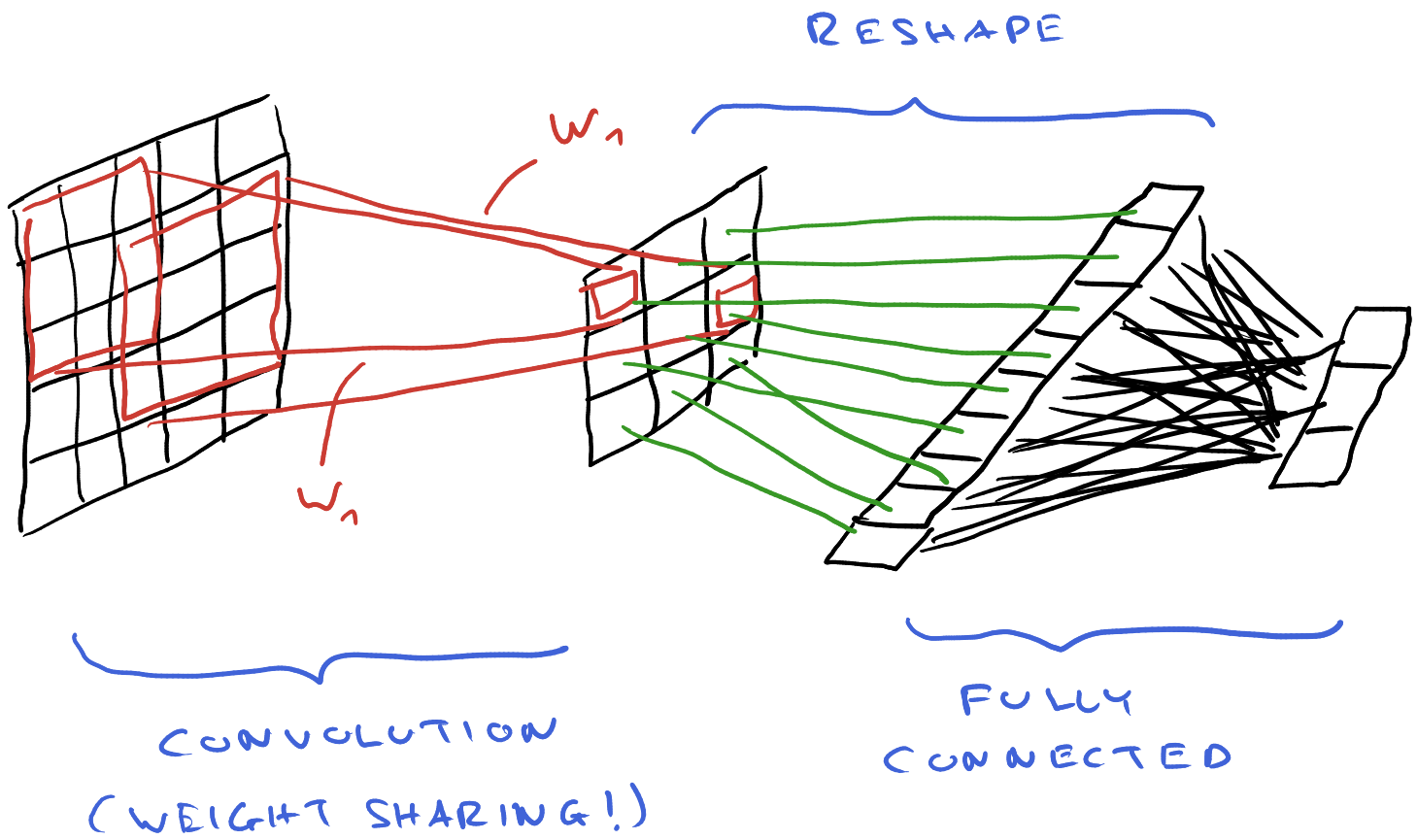


DROPOUT

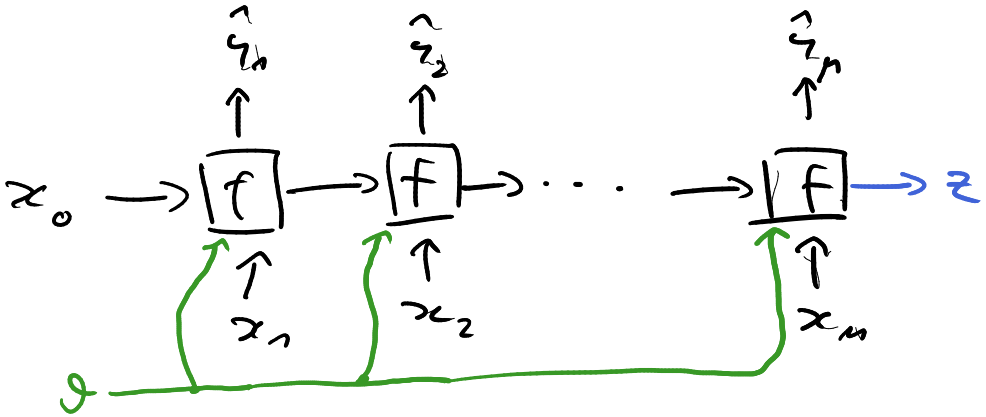


SPLIT LOSS

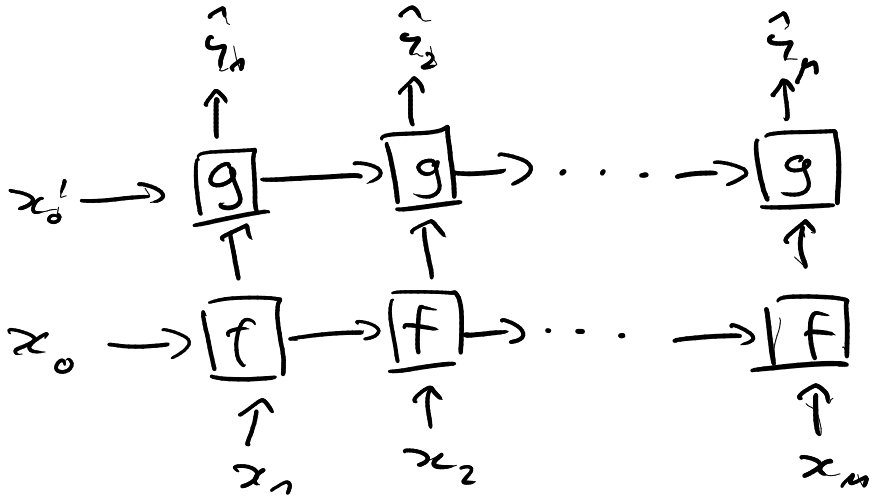
CONVOLUTIONAL NETWORKS

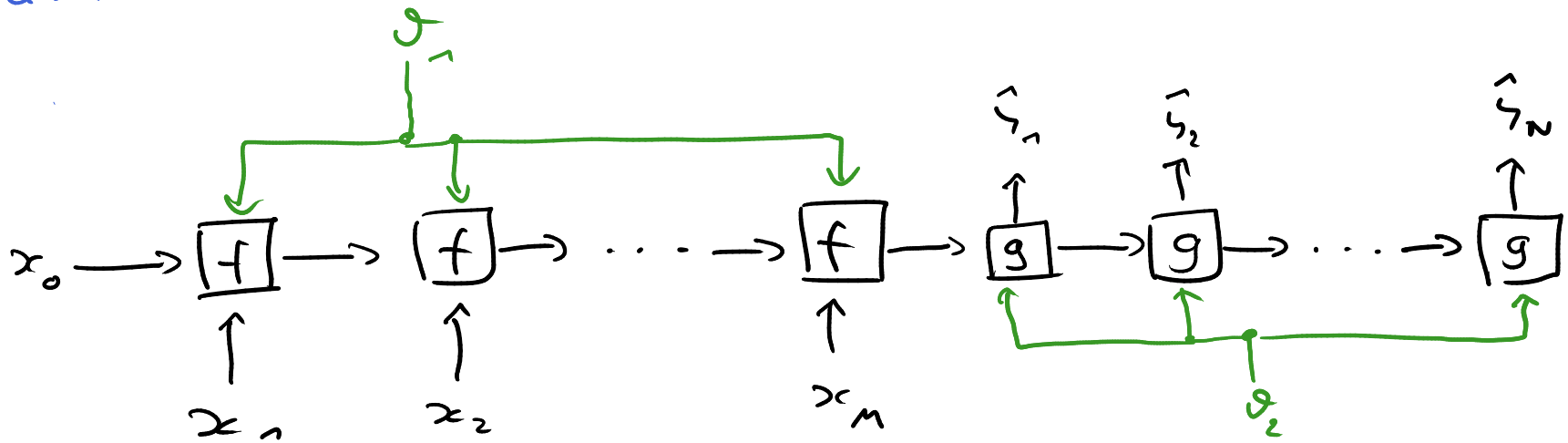


RECURRENT NETWORKS



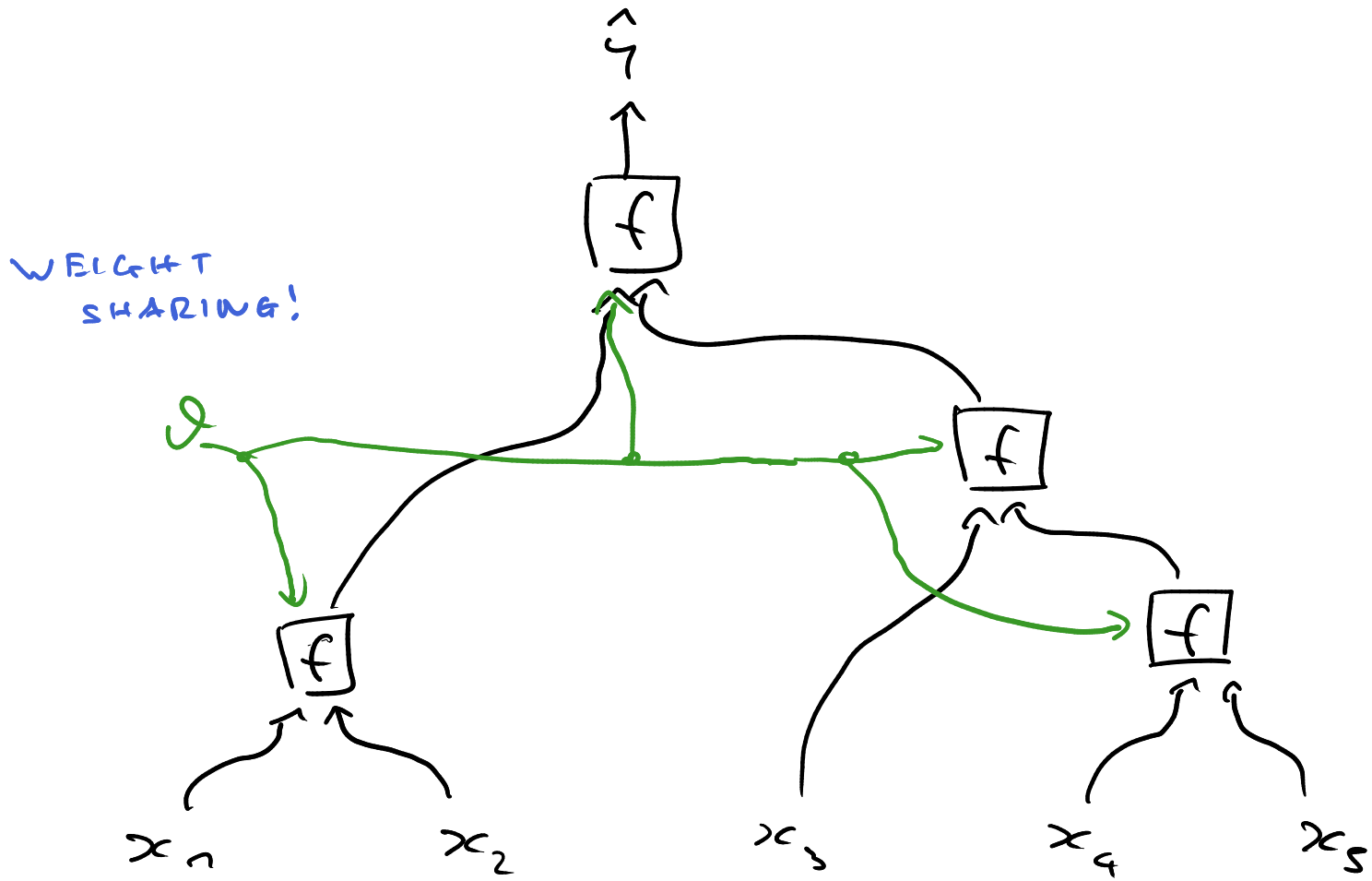
WEIGHT SHARING!



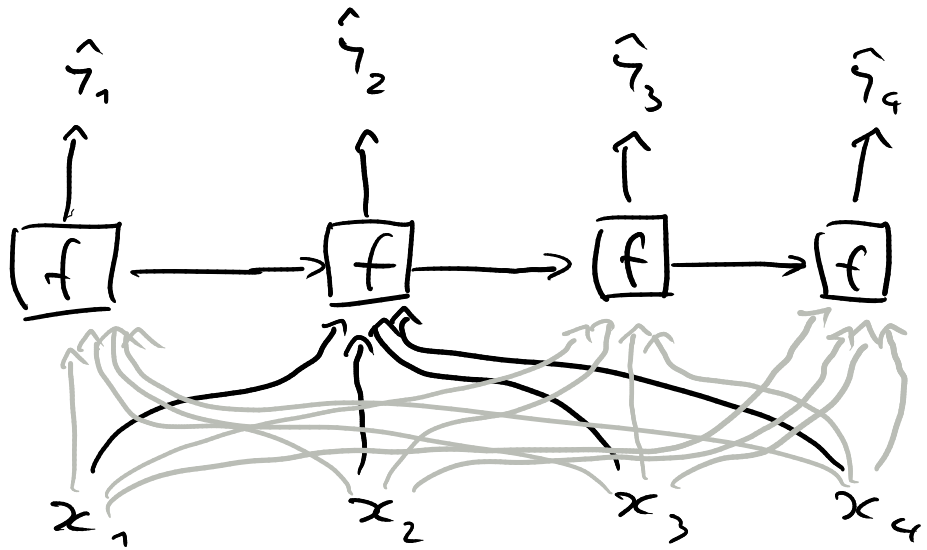


SEQ 2 SEQ

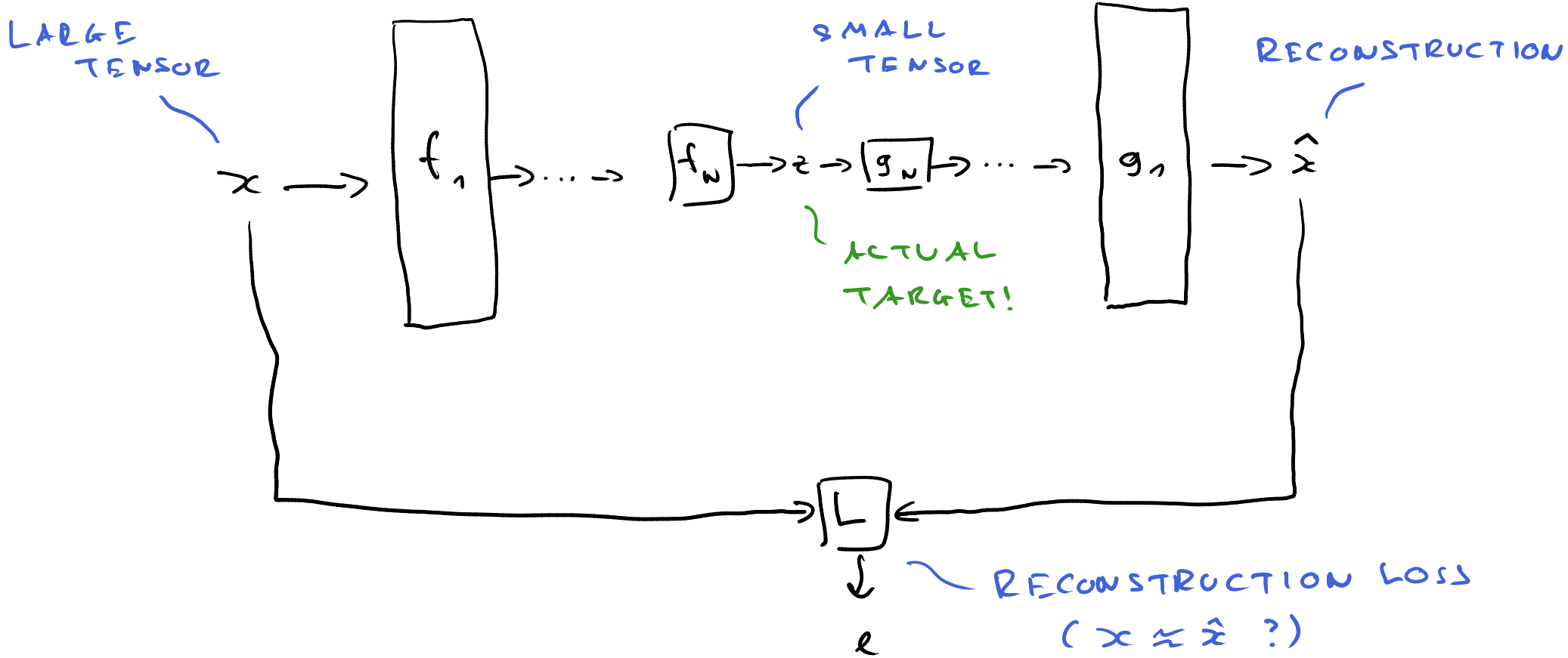
RECURSIVE NETWORKS



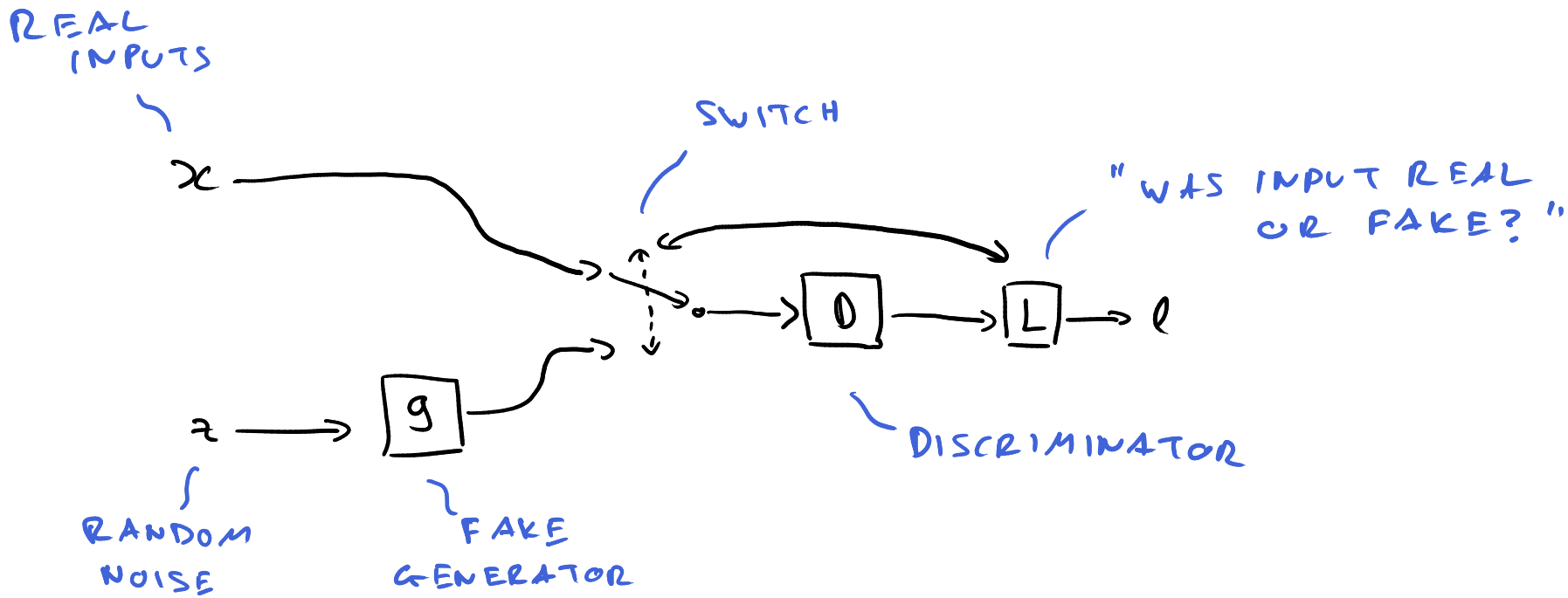
ATTENTION!



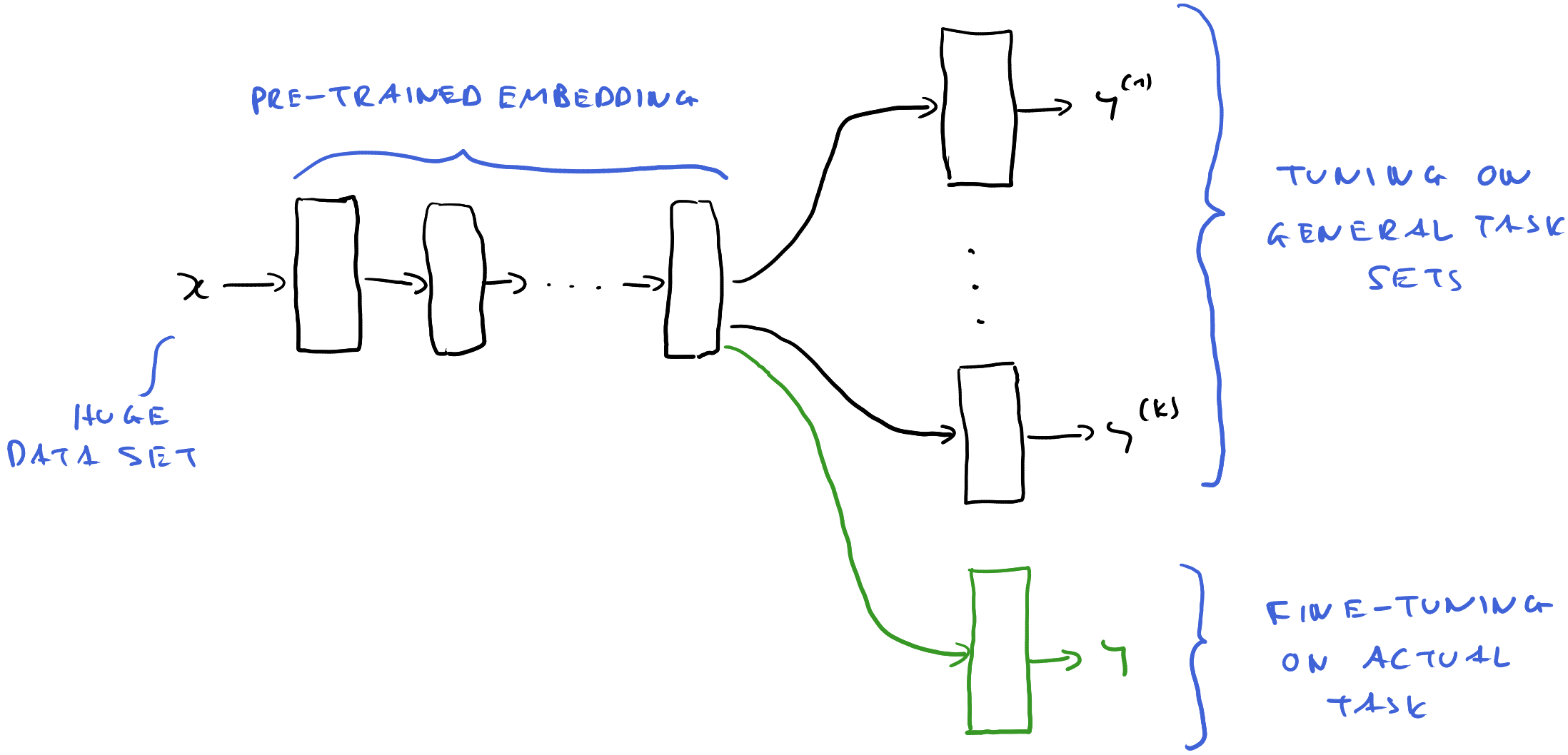
AUTOENCODER



GENERATIVE ADVERSERIAL NETWORKS



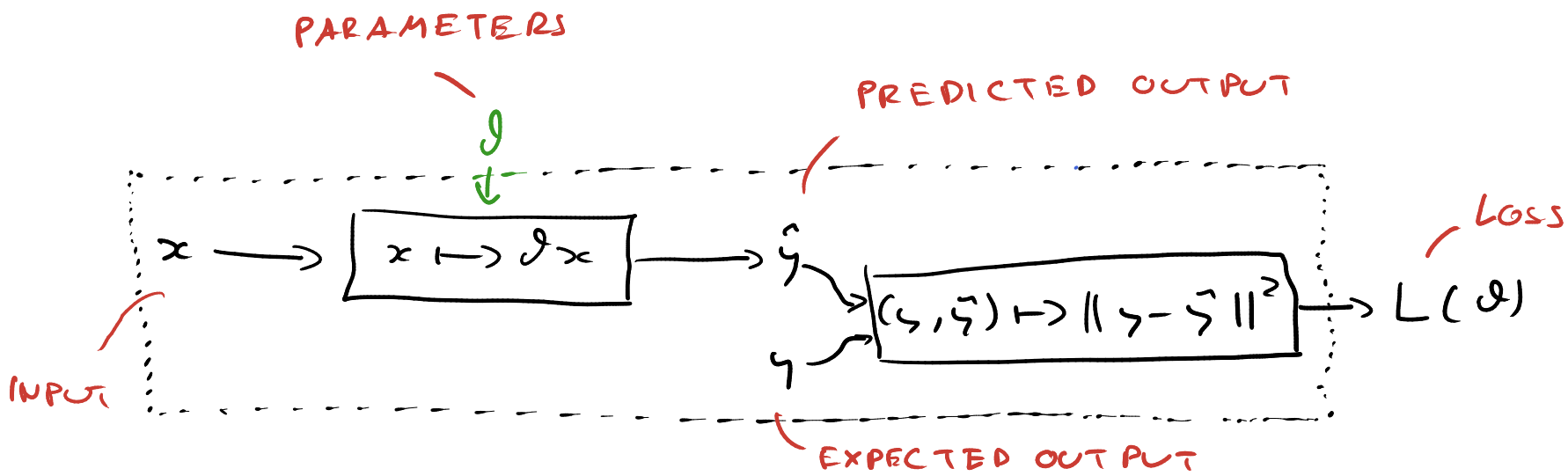
TRANSFER LEARNING



How are solutions implemented?

https://pxhere.com/en/photo/942155

OPTIMIZATION IN OLS



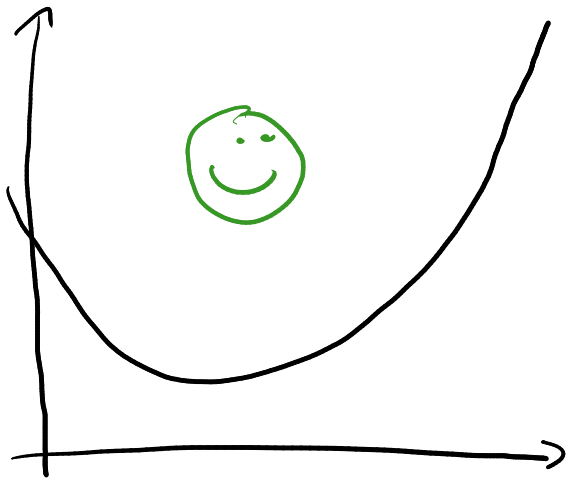
$$\frac{dL}{d\vartheta} \doteq \emptyset$$

"ABLEITEN, NULLSETZEN"

$$\vartheta^* = (X^T X)^{-1} X^T y$$

COMPLEXITY OF LOSS FUNCTIONS

LEAST SQUARES: QUADRATIC



- CONVEX
- SIMPLE MATH

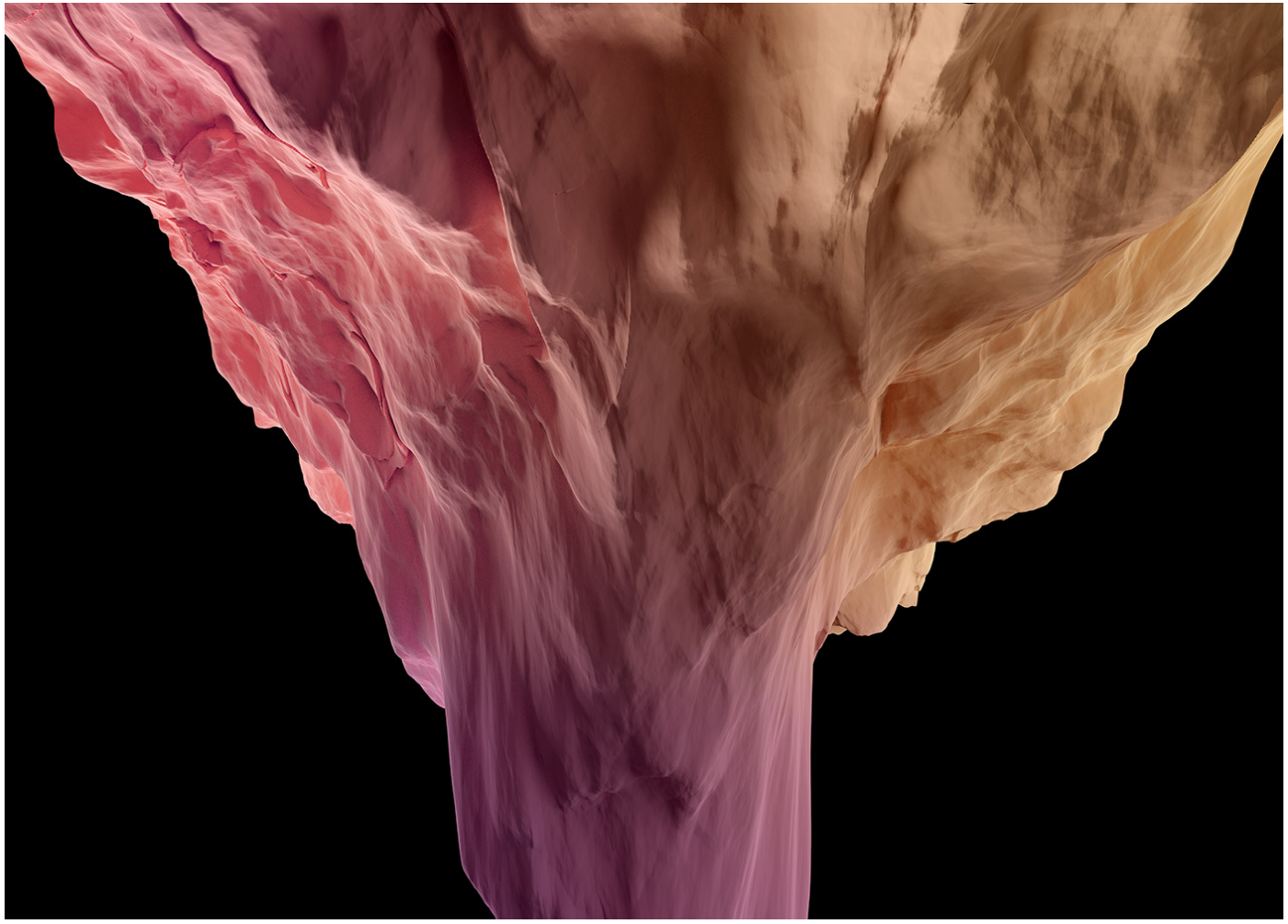
ANN: "LANDSCAPE"

- NON-CONVEX
- NO ANALYTICAL SOLUTION

Real Loss Landscapes

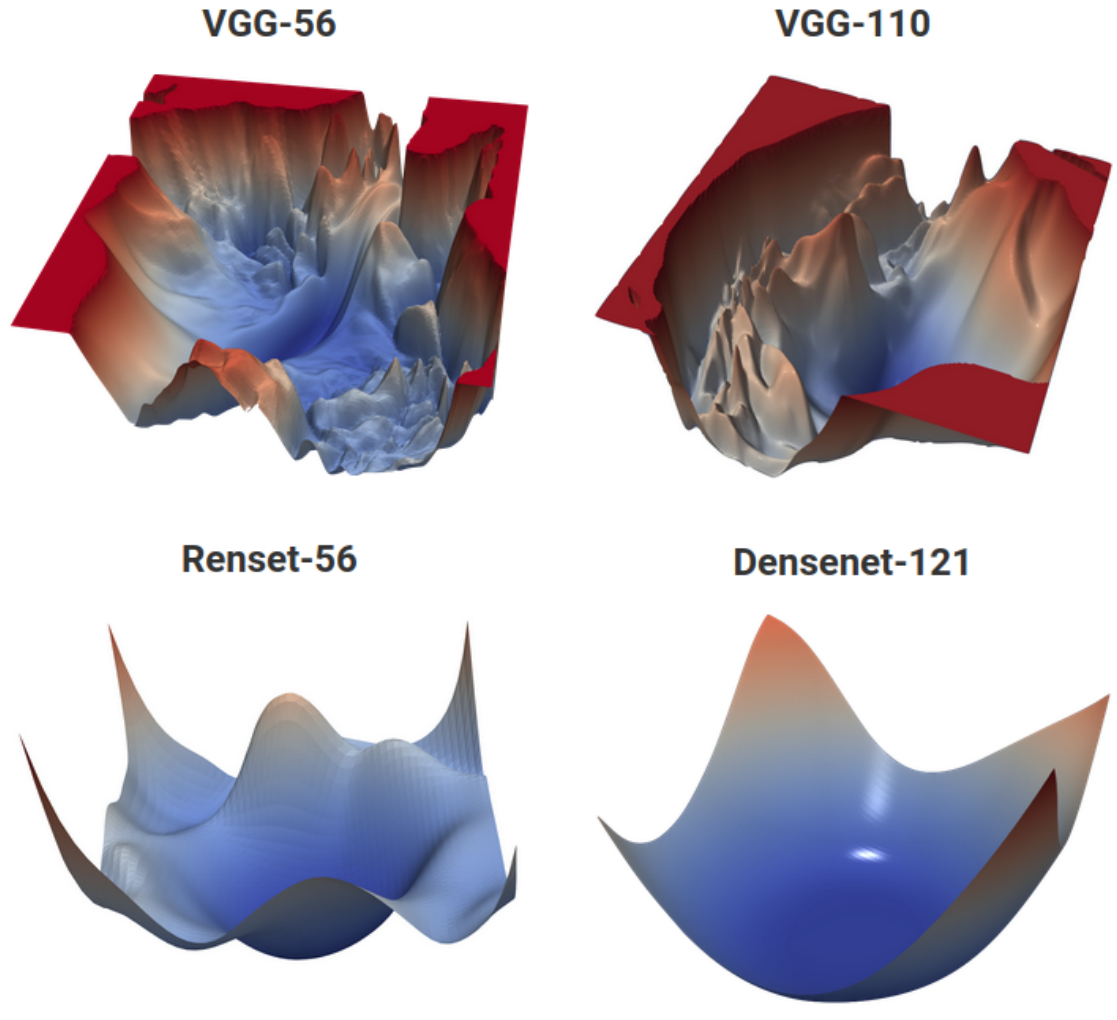
https://losslandscape.com/gallery/

THERE'S HOPE!



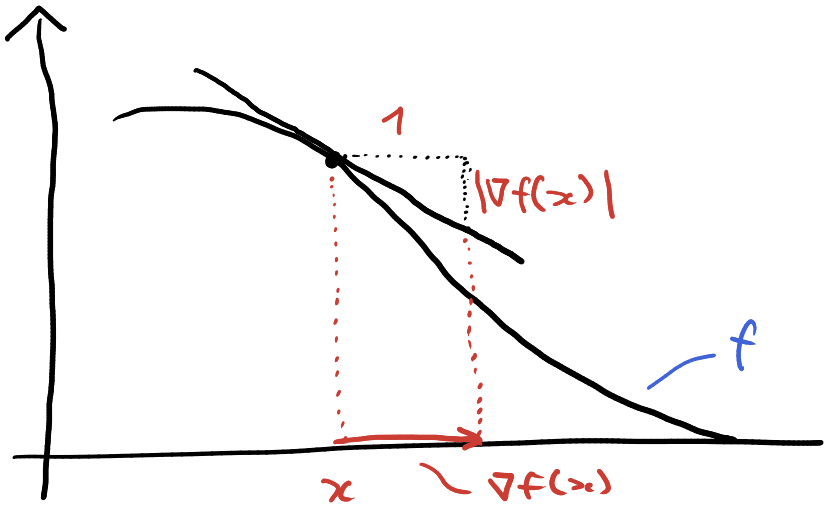
https://losslandscape.com/gallery/

INFLUENCE OF ARCHITECTURES

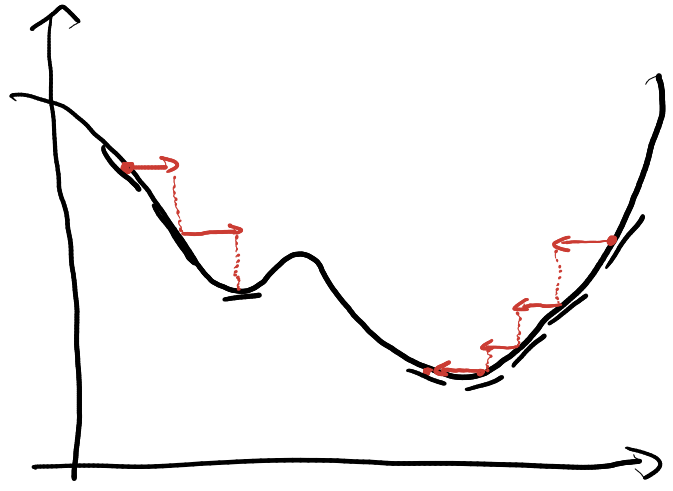


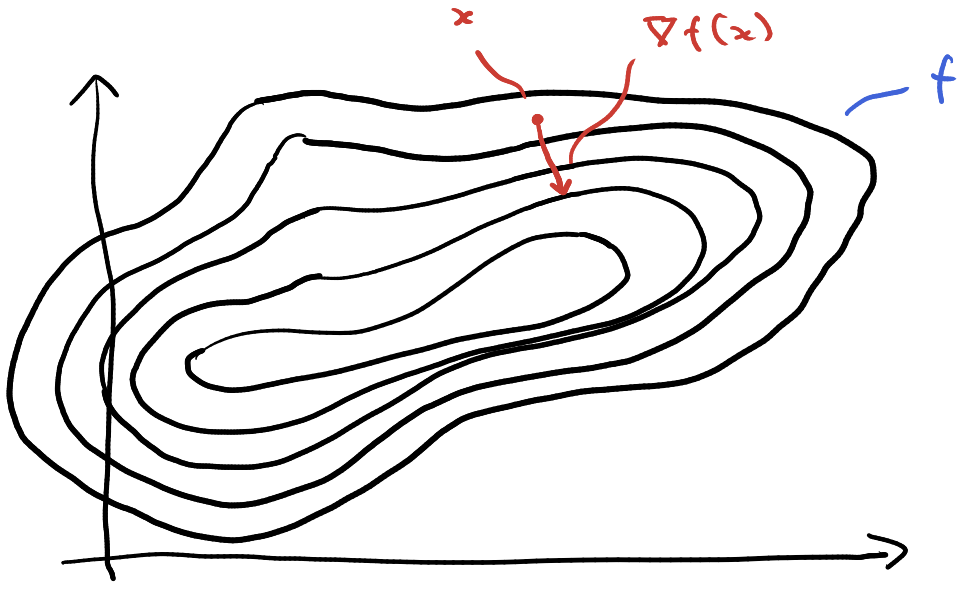
https://www.cs.umd.edu/~tomg/projects/landscapes/

GRADIENT DESCENT

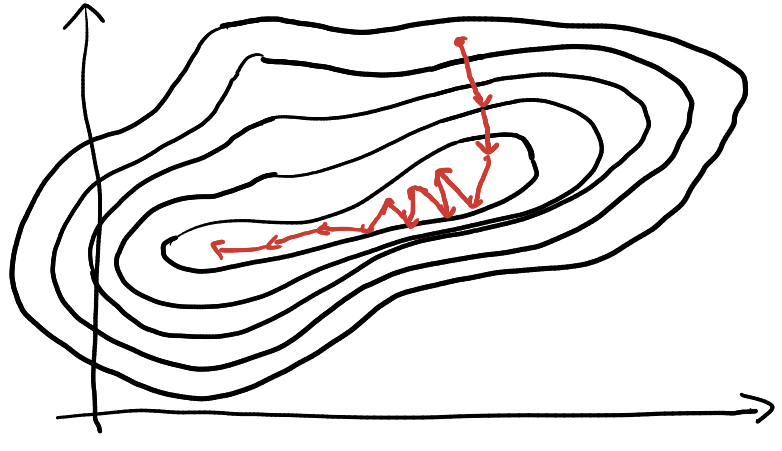


1D

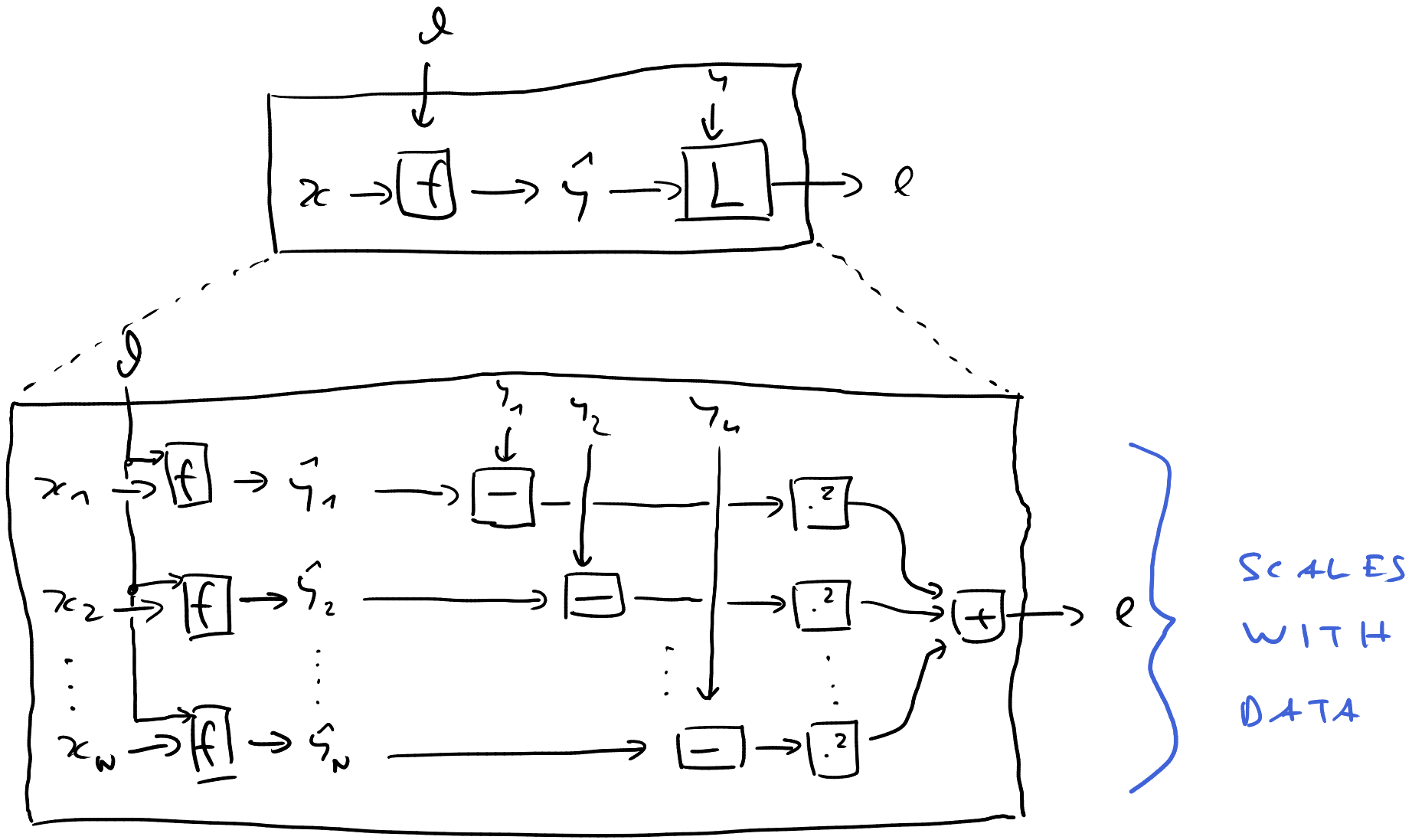




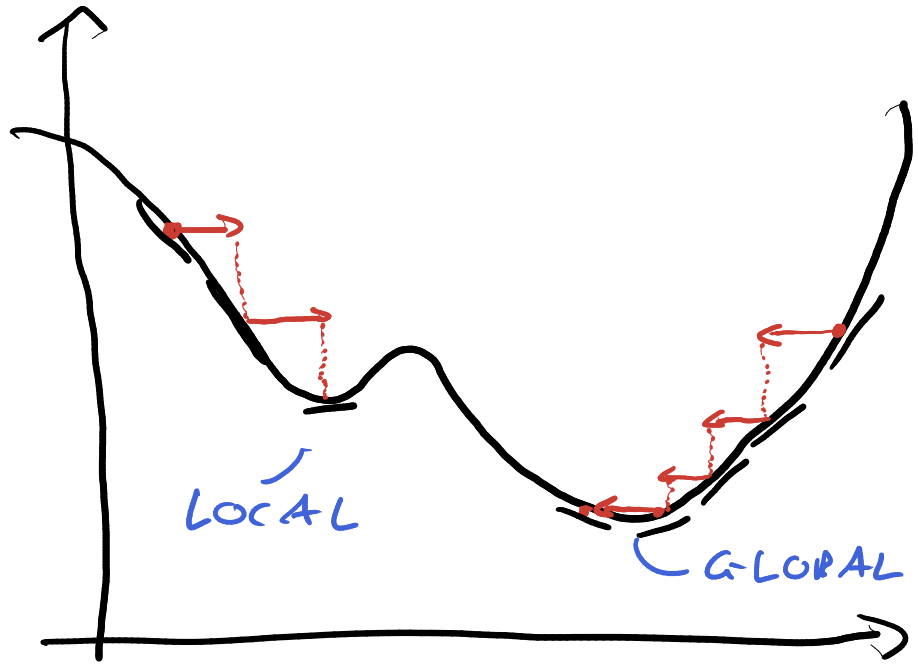
2D



NOT SO SIMPLE...



STUCK IN LOCAL OPTIMA



STOCHASTIC GRADIENT DESCENT!

1) SAMPLE $(\tilde{x}, \tilde{y})$ FROM DATA (SMALL!)

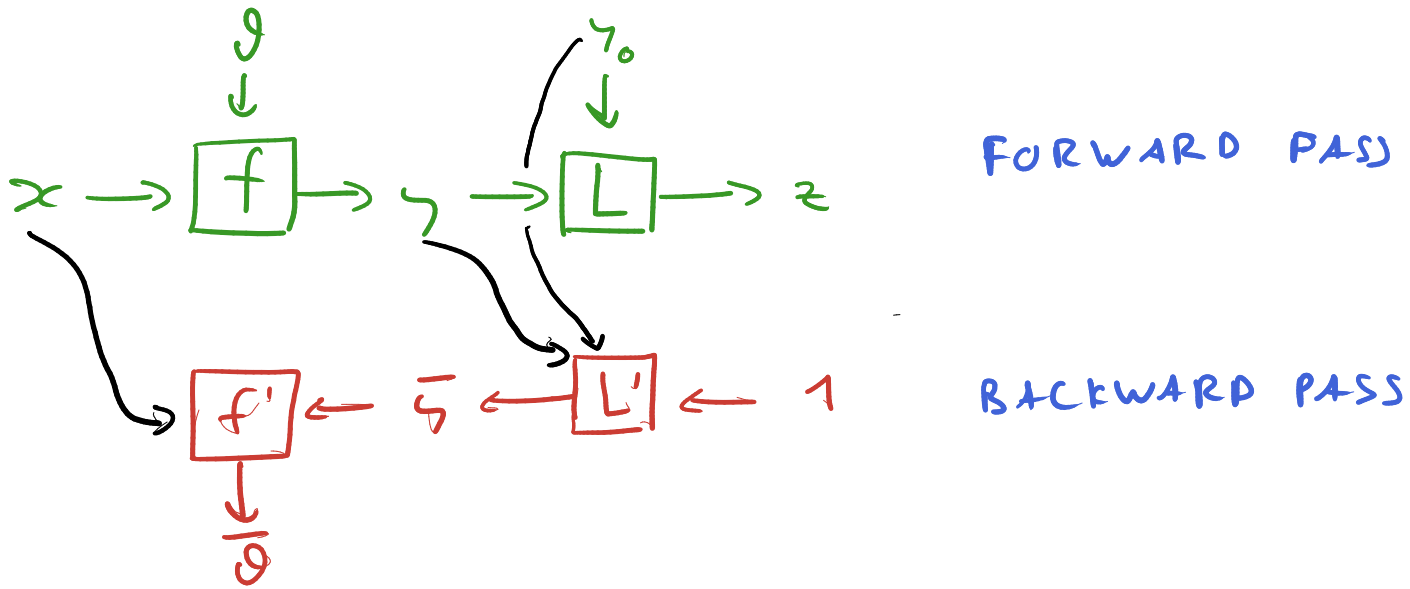
2) CALCULATE $\nabla L(\theta)$ ON SAMPLE ← FIXED SIZE!

3) DO GRADIENT STEP IN NOISY DIRECTION!

← WIGGLE OUT OF LOCAL OPTIMA!

→ ALSO: MORE ROBUST LEARNING

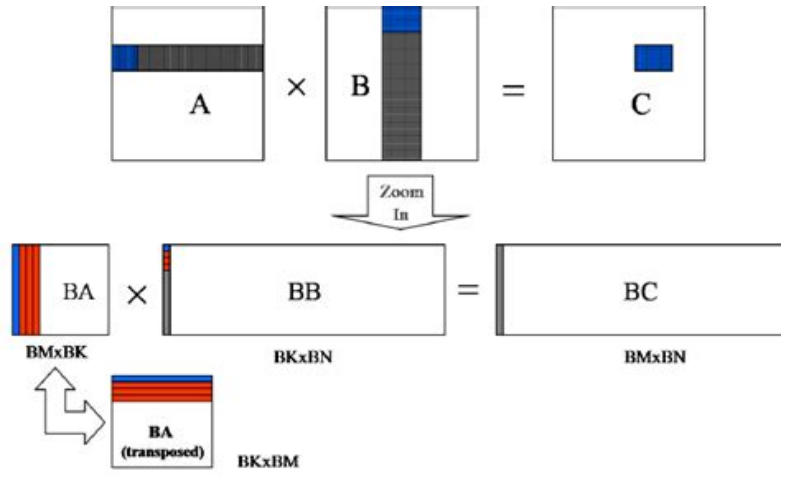
MAY THE GRAPH BE WITH YOU



$$\bar{\theta} = \nabla_\theta L(\overbrace{f(x,\theta)}^{y}, y_0) = \underbrace{L'(\overbrace{f(x,\theta)}^{y}, y_0)}_{\bar{y}} f'(x,\theta)$$

Linear Algebra's Dirty Secrets

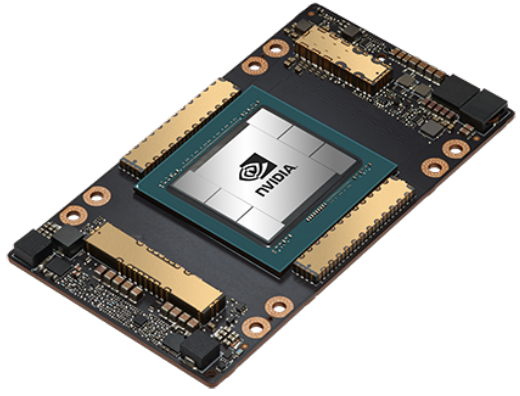
```
1  void Gemm(int m, int n, int k, double *A, double *B, double *C) {
2    // Declarations: mc, nc, kc,...
3    for ( jc = 0; jc < n; jc += nc )                          // Loop 1
4      for ( pc = 0; pc < k; pc += kc ) {                      // Loop 2
5        // B(pc : pc + kc − 1, jc : jc + nc − 1) → Bc
6        Pack_buffer_B(kc, nc, &B(pc,jc), &Bc);
7        for ( ic = 0; ic < m; ic += mc ) {                    // Loop 3
8          // A(ic : ic + mc − 1, pc : pc + kc − 1) → Ac
9          Pack_buffer_A(mc, kc, &A(ic,pc), &Ac);
10         // Macro-kernel:
11         for ( jr = 0; jr < nc; jr += nr )                   // Loop 4
12           for ( ir = 0; ir < mc; ir += mr ) {               // Loop 5
13             // Micro-kernel:
14             //     Cc(ir : ir + mr − 1, jr : jr + nr − 1) +=
15             //         Ac(ir : ir + mr − 1, 1 : 1 + kc − 1) ·
16             //         Bc(j, 1 : 1 + kc − 1, r : jr + nr − 1)
17             Gemm_mkernel( mr, nr, kc, &Ac(ir,1), &Bc(1,jr),
18                                                  &Cc(ir,jr) );
19           }
20       }
21     }
22 }
```



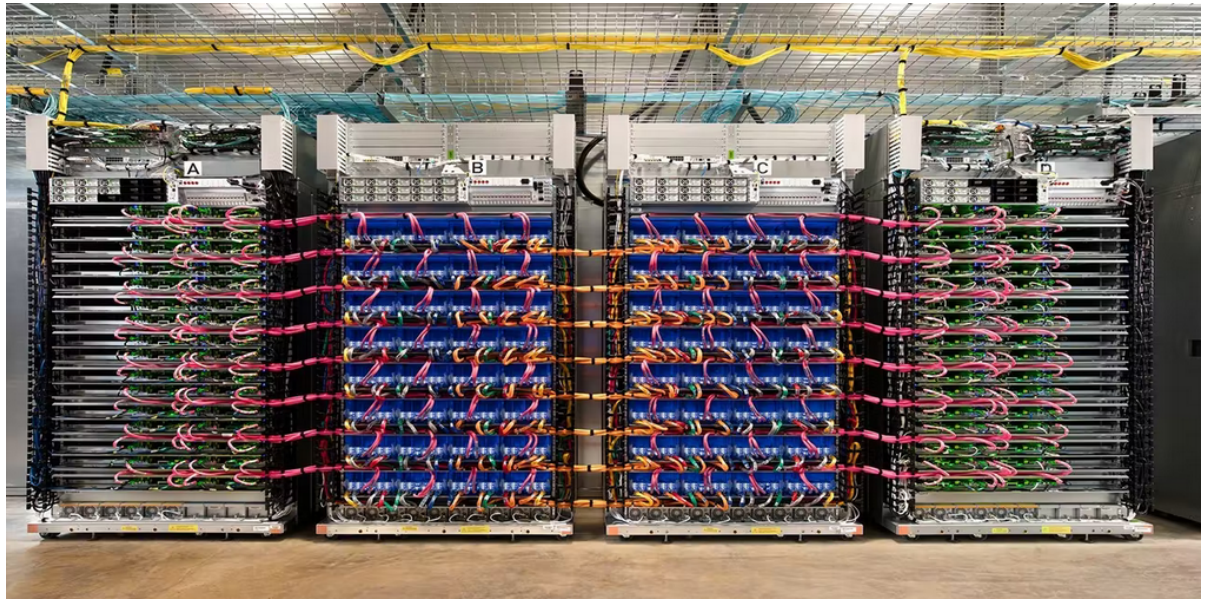
http://dx.doi.org/10.1007/s10586-019-02927-z

http://dx.doi.org/10.1007/978-3-642-01970-8_89

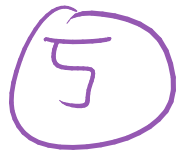
MAKE THE HARDWARE GREAT AGAIN



https://www.gigabyte.com/id/Solutions/nvidia-a100

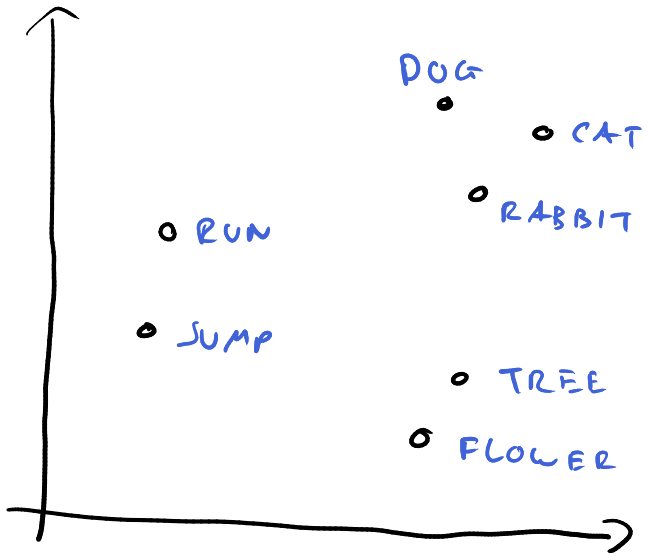


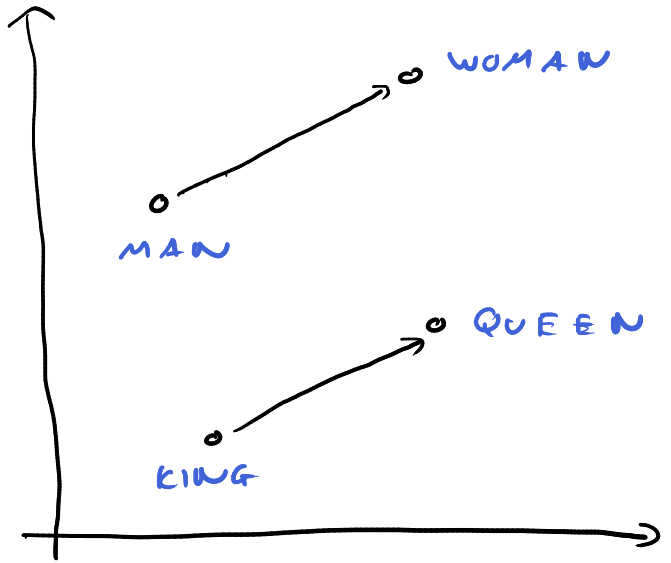
https://www.inverse.com/article/31745-google-tensor-processing-unit



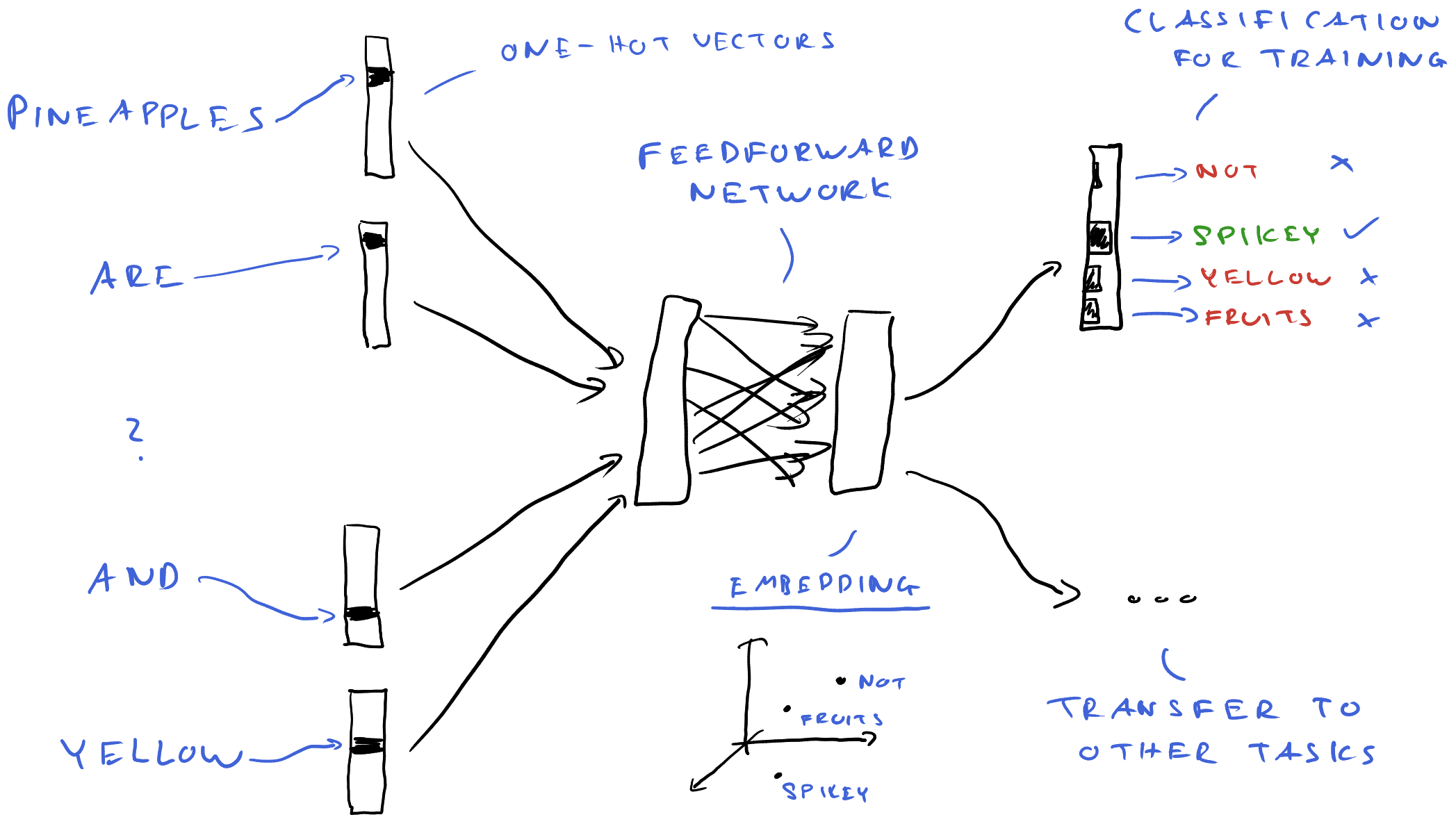
SOME LANGUAGE-RELATED USE CASES

WORD EMBEDDINGS

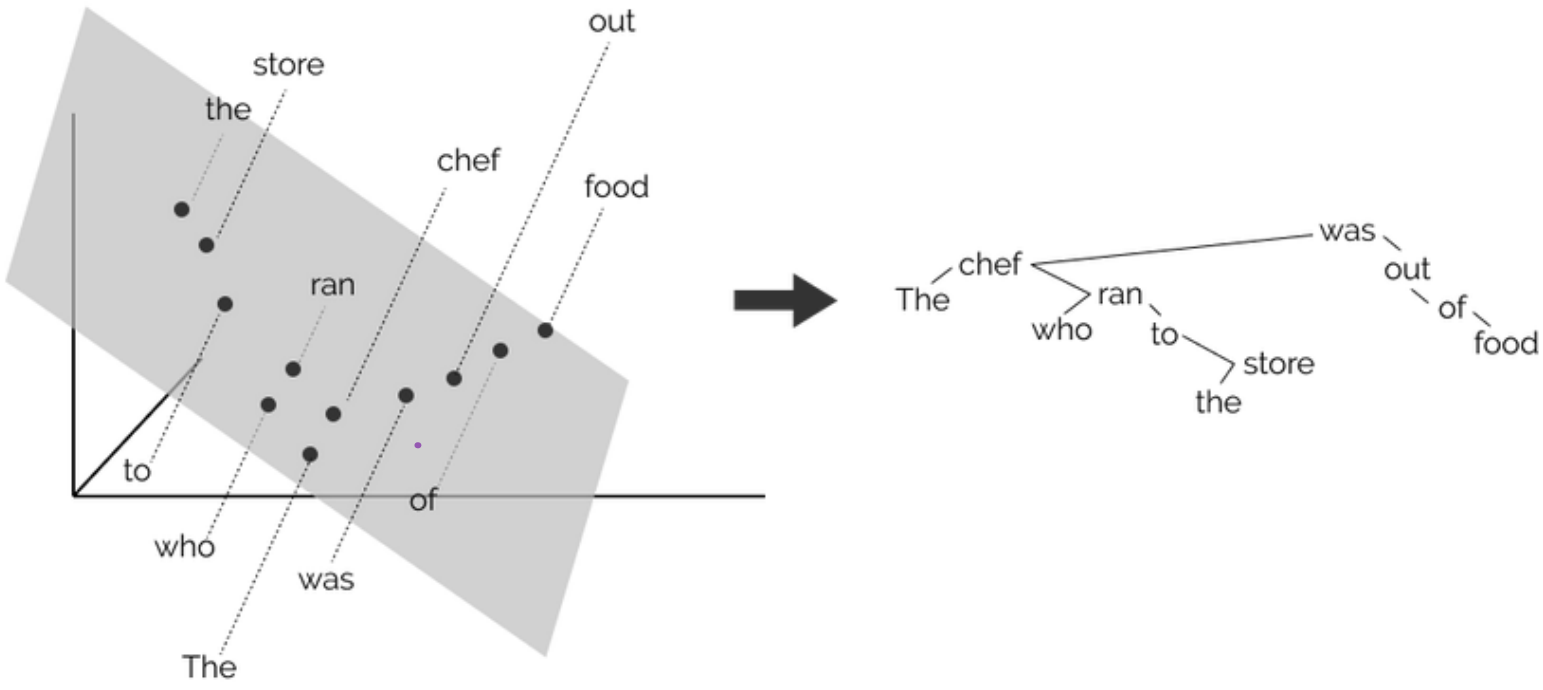


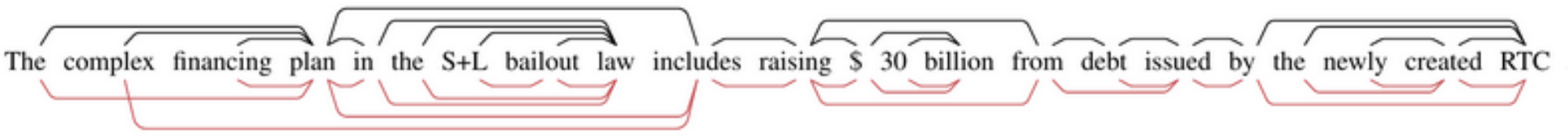


CONTINUOUS BAG-OF-WORDS



SYNTAX FROM EMBEDDINGS

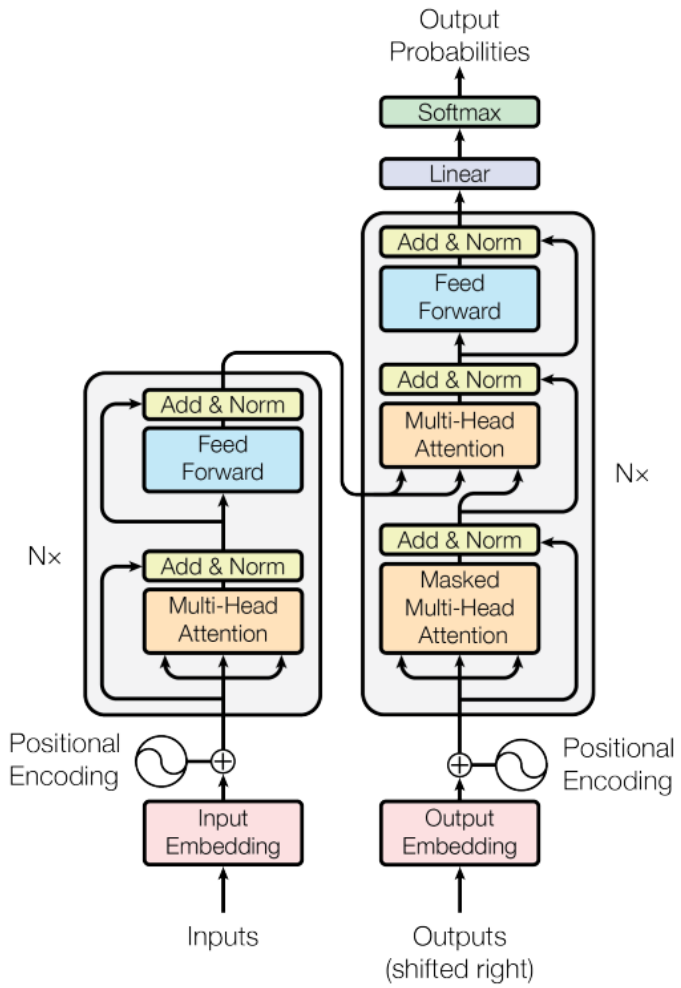




https://nlp.stanford.edu/~johnhew/structural-probe.html

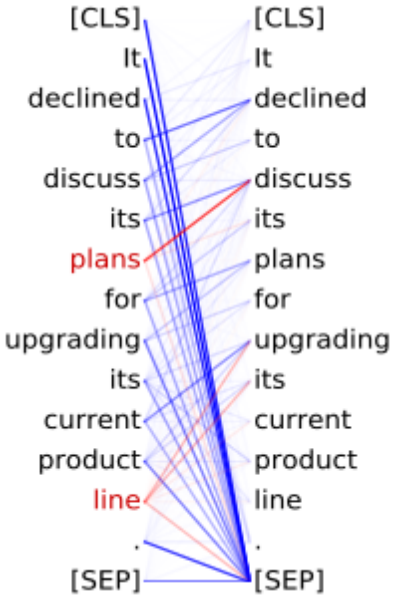
ATTENTION IS ALL YOU NEED

TRANSFORMER:



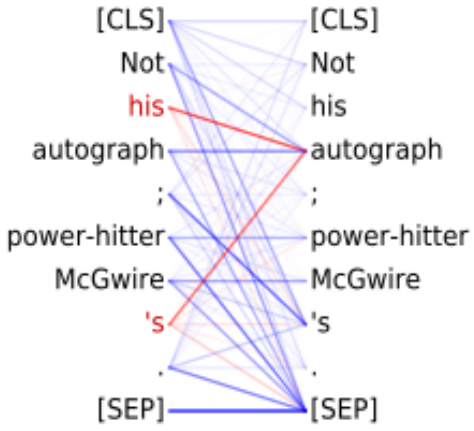
https://dl.acm.org/doi/10.5555/3295222.3295349

WHAT DOES ATTENTION LOOK AT?



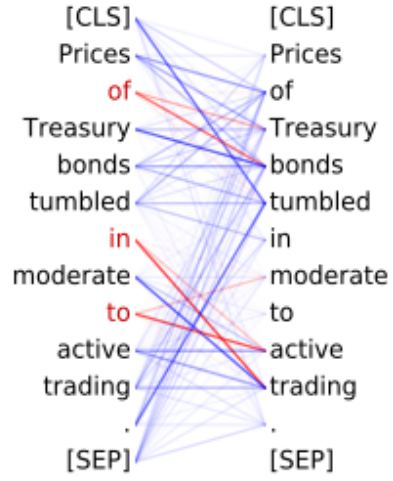
DOBJ

86.8%



POSS

80.5%



POBJ

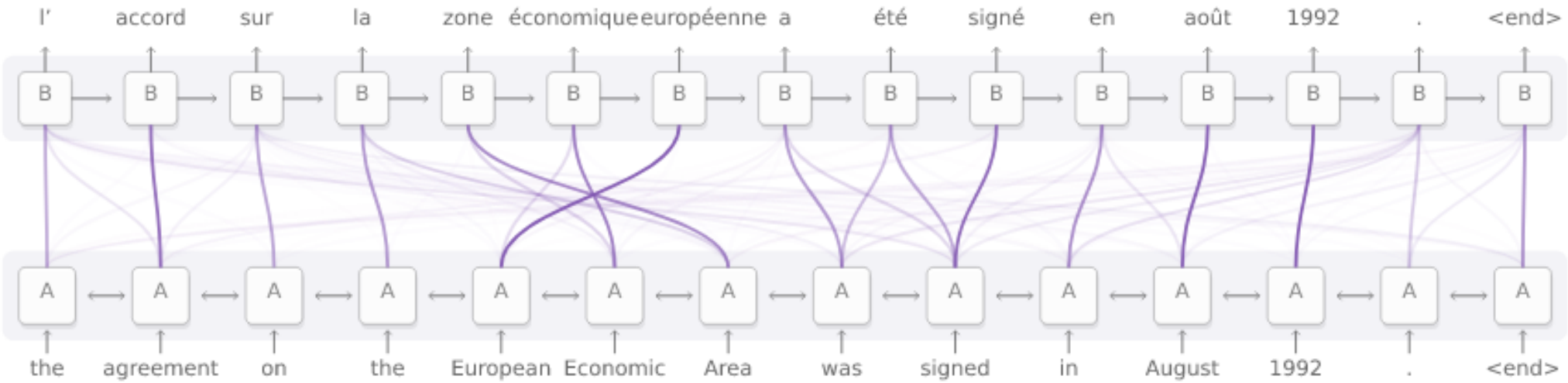
76.3%

COREFERENCE

65.1%

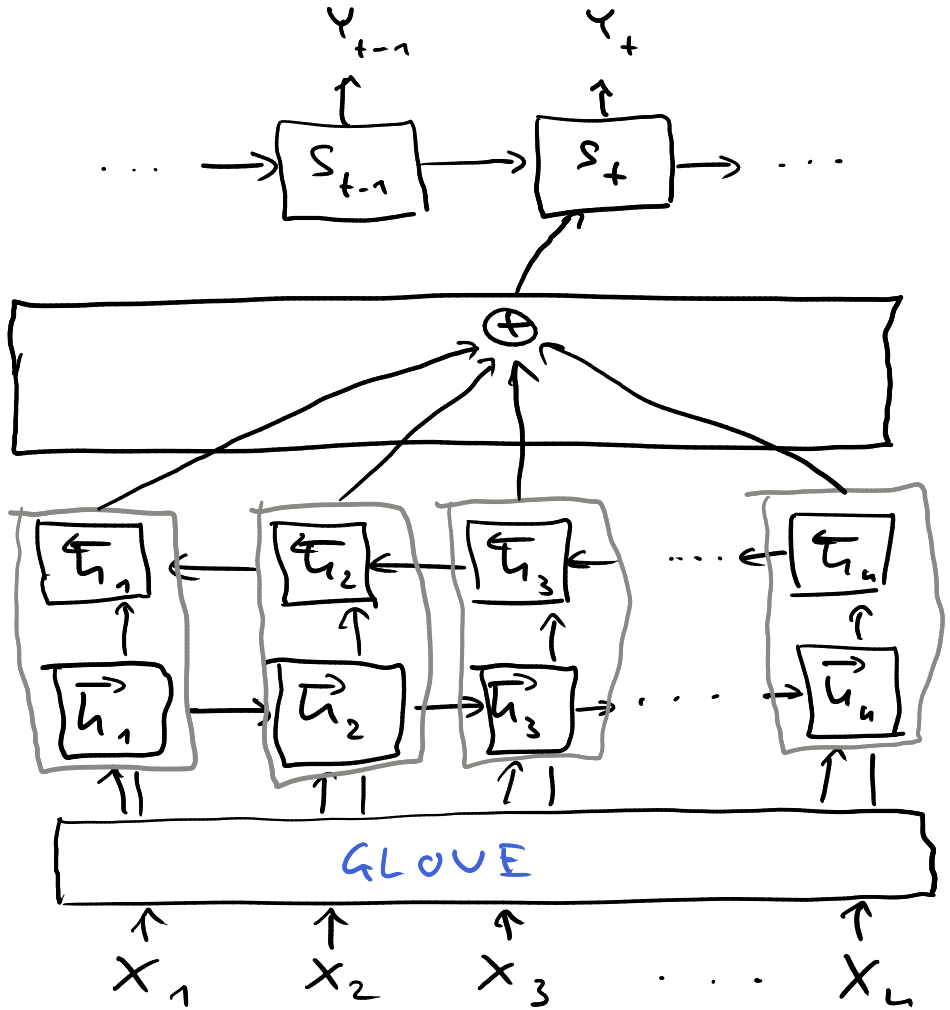
https://aclanthology.org/W19-4828/

Attention for Sequence Prediction



distill.pub/2016/augmented-rnns/

TYPICAL ARCHITECTURE



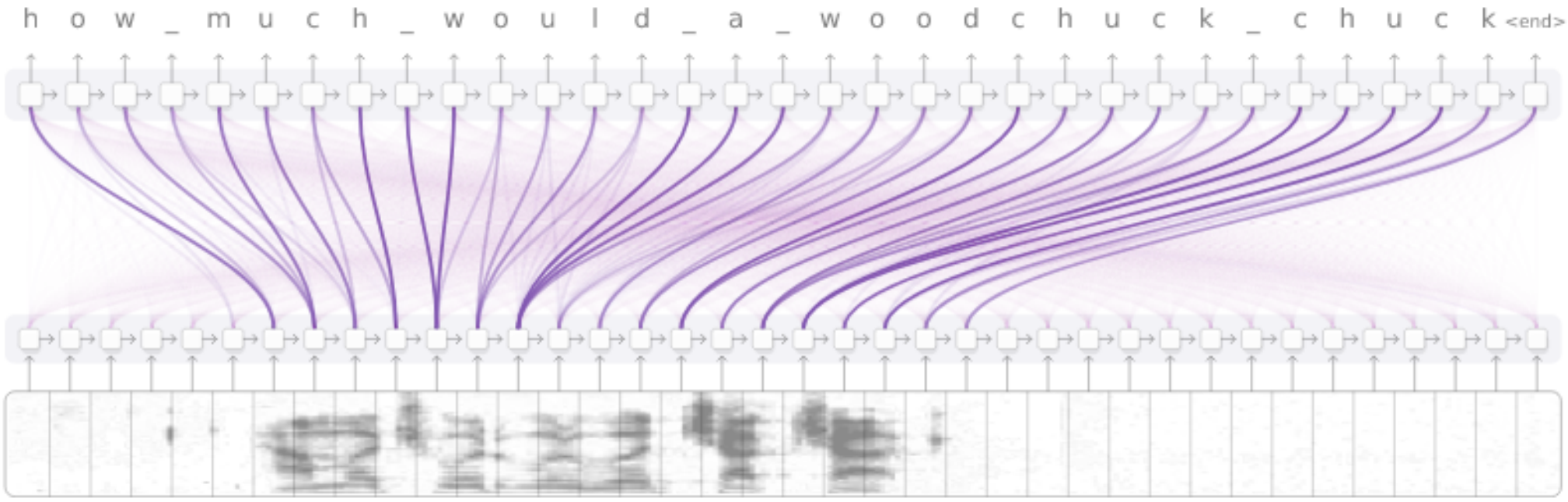
DECODER RNN

ATTENTION

BIDIRECTIONAL RNN

EMBEDDING

WAV2VEC



distill.pub/2016/augmented-rnns/

DIALOGUE SYSTEMS

https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html

References

Introductory:

- Buckner, C. (2019). Deep learning: A philosophical introduction. Philosophy Compass, 14(10), e12625. https://doi.org/10.1111/phc3.12625
- Bzdok, D., Altman, N., & Krzywinski, M. (2018). Statistics versus machine learning. Nature Methods, 15(4), 233.
- Liquet, B., Moka, S., Nazarathy, Y. (2022). The Mathematical Engineering of Deep Learning (draft): Introduction. https://deeplearningmath.org/drafts/chap1.pdf

General:

- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Murphy, K. P. (2012). Machine learning: A probabilistic perspective. MIT Press.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org

References

Optimization & Learning:

- Kochenderfer, M. J., & Wheeler, T. A. (2019). Algorithms for Optimization. MIT Press. https://algorithmsbook.com/optimization/
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. J. (2006). A Tutorial on Energy-Based Learning. http://yann.lecun.org/exdb/publis/pdf/lecun-06.pdf