# Singular Value Decomposition and Neural Networks

Bernhard Bermeitinger [1,2] ✉ • Tomas Hrycej [1] • Siegfried Handschuh [1,2]

{bernhard.bermeitinger,tomas.hrycej,siegfried.handschuh}@unisg.ch

[1] Institute for Computer Science, Universität St.Gallen, Switzerland [2] FIM, Universität Passau, Germany

## Singular Value Decomposition

- SVD constitutes a bridge between *Linear Algebra* and *Multi-Layer Neural Networks*.
- SVD results in a good initial guess for weight and bias parameters.

It is a decomposition of an arbitrary matrix $A$ of size $m \times n$ into three factors:

$$A = USV^T$$

where $U$ and $V$ are orthonormal and $S$ is of identical size as $A$, consisting of a diagonal matrix $D_0$ and a zero matrix.

SVD is then simplified to:

$$A = USV^T = U\,[S_0, 0]\,[V_0, V_x]^T = US_0V_0^T$$

## Motivation

A further application of SVD is an explicit formula for a matrix pseudo-inverse. Pseudo-inverse $A^+$ is the analogy of an inverse matrix for the case of non-square matrices, with the property

$$AA^+A = A$$

It can be easily computed with the help of SVD:

$$X^+ = VS_{\text{inv}}^T U^T$$

**Over-Determined:** For over-determined problems with input $X$ and output $Y$, the least-square-optimum solution is the linear regression with matrix $B$:

$$y = Bx + a$$

**Under-Determined:** For under-determined problems, like computer vision or corpus-based semantics, the number of training examples is substantially lower than the dimensions of the input. The projection operation for input $X$ and output $Y$ is then given as:

$$\hat{x} = XX^+x = X\left(X^TX\right)^{-1}X^Tx$$

## SVD and Linear Networks

*Linear* Networks with one hidden layer of size $p$ can represent the best linear mapping from input $x$ to output $y$: $y = Bx$
The best approximation with rank limitation to $\hat{r}$ is:

$$y = U_1S_1V_1^Tx$$

## SVD and Initializing Nonlinear Neural Networks

The nonlinear activation function *sigmoid*

$$s(x) = \frac{1}{1+e^{-x}}$$

is nearly linear around $x = 0$ with its derivative 0.25. By rescaling this unit to

$$f(x) = 2s(x) - 1 = \frac{2}{1+e^{-2x}} - 1$$

it becomes a nonlinear function which is unity around $x = 0$.

- A Neural Network with one hidden layer using *sigmoid* **behaves like a linear network for small activation values of the hidden layer**.

## Computing Experiments

| Type | #input | #output | #hidden | #training | #parameters | #constraints |
|---|---|---|---|---|---|---|
| A | 100 | 50 | 20 | 80 | 3070 | 4000 |
| B | 300 | 150 | 60 | 240 | 27 210 | 36 000 |
| C | 1000 | 500 | 200 | 800 | 300 700 | 400 000 |

| Algorithm | Init. | size class $A$ | | size class $B$ | | size class $C$ | |
|---|---|---|---|---|---|---|---|
| | | #iter. | $F_{opt} \times 10^{-3}$ | #iter. | $F_{opt} \times 10^{-3}$ | #iter. | $F_{opt} \times 10^{-3}$ |
| SVD | — | — | 10.200 | — | 10.559 | — | 10.814 |
| SGD | Random | 2000 | 30.361 | 2000 | 90.402 | 2000 | 177.095 |
| RMSprop | Random | 2000 | 0.040 | 2000 | 0.096 | 2000 | 0.260 |
| Adadelta | Random | 2000 | 1.290 | 2000 | 6.748 | 2000 | 31.076 |
| CG | Random | 637 | 0.002 | 821 | 0.012 | — | — |
| SGD | SVD | 2000 | 1.779 | 2000 | 4.145 | 2000 | 7.254 |
| RMSprop | SVD | 2000 | 0.030 | 2000 | 0.085 | 2000 | 0.248 |
| Adadelta | SVD | 2000 | 0.062 | 2000 | 0.511 | 2000 | 2.086 |
| CG | SVD | 316 | 0.000 | 233 | 0.021 | — | — |

## Conclusion

SVD can be used as a good initial guess for the network parameters. **The quality of this initial guess may be better than weakly performing (but widely used) methods such as SGD ever reach.**

## Future Work

- Apply in large datasets (MNIST, CIFAR, etc.)
- Directly implement in *TensorFlow*: Custom initialization using SVD and Conjugate Gradient Optimization

Universität St.Gallen

UNIVERSITÄT PASSAU

Theoretical Neural Computation, Information and Optimization, Deep Learning

ICANN19

28th International Conference on Artificial Neural Networks
17.–19. September 2019, Munich, Germany