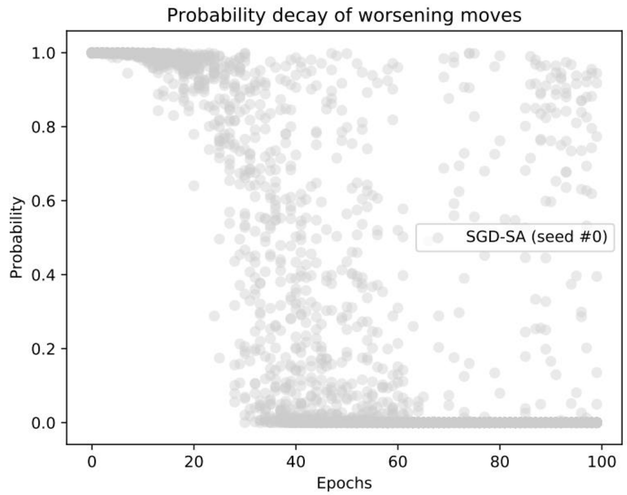
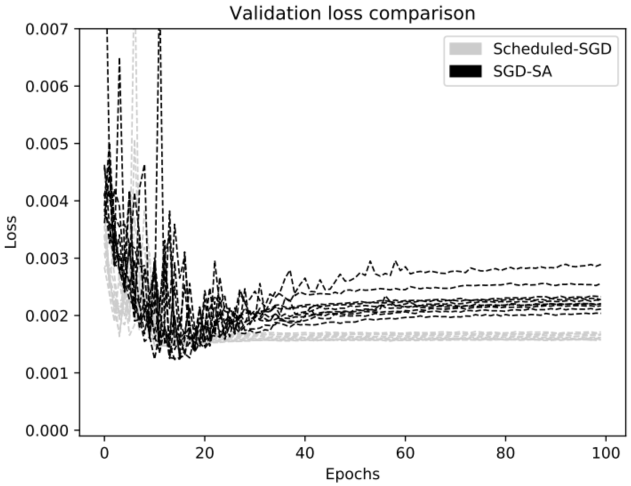
Embedding Simulated Annealing within Stochastic Gradient Descent

Matteo Fischetti and Matteo Stringher

University of Padova



(a) Probability of accepting worsening moves



(a) Validation loss

OLA 2021, Catania, 21 June 2021

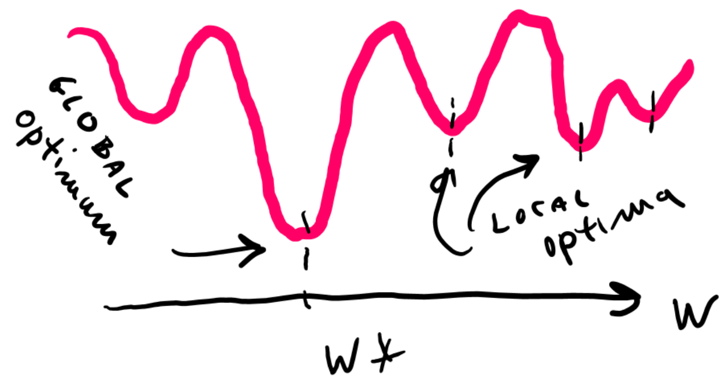
ML training as a (deterministic) optimization problem?

- Training in Machine Learning (ML)

  Given
  - A ML architecture (e.g., a DNN) with **weights** w_j
  - A **training set** T containing a possibly huge n. of sample points x^i
  - A nonnegative **loss function** $L_T(w)$ computed w.r.t. set T

  Find
  - A **global optimal solution** w* of the problem $\mathbf{min_w\ L_T(w)}$

  Classical optimization issues:
  - Large scale
  - non-convexity of $L_T(w)$,
  - local vs global optima, etc.



Training however is NOT a (deterministic) optimization problem

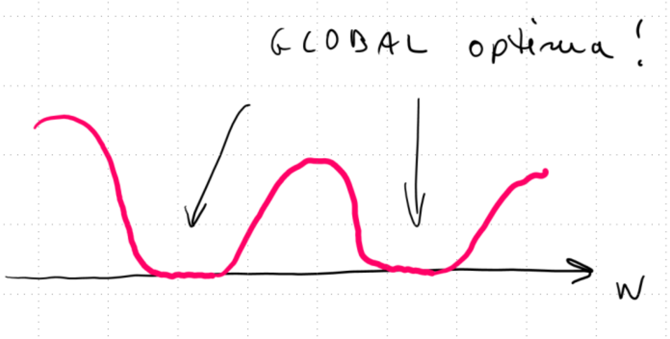
- Modern DNNs are usually highly over-parametrized!

Table 1 showing different architectures statistics

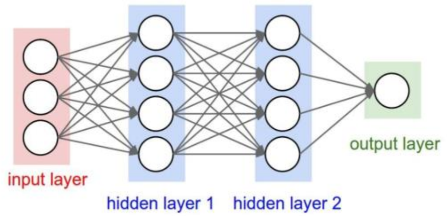
Model	AlexNet	GoogleNet	ResNet152	VGGNet16	NIN
#Param	60M	7M	60M	138M	7.6M
#OP	1140M	1600M	11300M	15740M	1100M
Storage (MB)	217	51	230	512.24	29

- → $\mathbf{\min_w L_T(w) = 0}$ (and, **by design**, quite easy to solve)

- Many GLOBAL optimal solutions w* with $\mathbf{L_T(w^*) = 0}$ exist!
- Although perfect on the training set, these solutions are NOT equivalent in terms of **generalization** (i.e., performance on unseen data)



The real training problem



- Training in Deep Learning

  Given
  - A DNN architecture with **weights** w_j
  - A **training set** T containing a possibly huge n. of sample points x^i
  - A **validation set V** containing a large n. of verification points x^i
  - A nonnegative **loss function** $L_S(w)$ w.r.t. a set of points S

  Find
  - A **sequence** of solutions w with $\mathbf{L_T(w) \approx 0}$ and choose among them a sol. w^* such that $\mathbf{L_V(w^*)}$ is as small as possible

  Warning:
  - Validation points can only be used ***sporadically*** (no gradient information or alike can be used)

A «turbulent» SGD

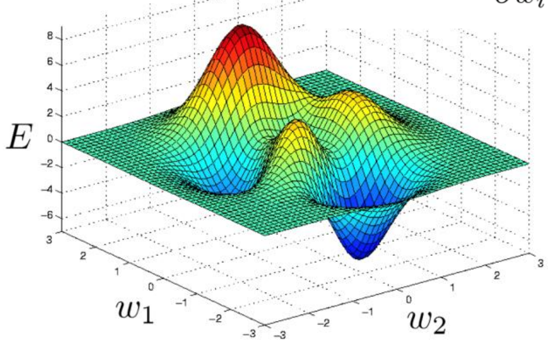
- **Key issue**: produce **diversified** optimal solutions over T, so as to have more freedom in picking the best w* w.r.t. the **validation set V**
  - Multi-start SGD: start with different random initial weights w
  - Hyper-parameter tuning
  - Change hyper-parameters in a cyclic way within the SGD run
  - …

- Our goal: produce a «**more turbulent**» sequence of solutions that lead to **more variation on the validation set** (even if this can slow down convergence on the training set)

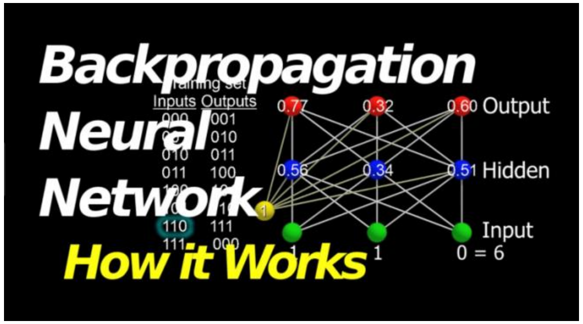
- We propose to modify the classical SGD alg. by implementing a **step-rejection test** in the vein of **Simulated Annealing (SA)**

We build on the three pillars of (practical) Deep Learning

$$w_i' = w_i - \alpha \frac{\partial E}{\partial w_i}$$



1) Stochastic Gradient Descent



2) Backpropagation

3) GPUs (and open-source Python libraries like Keras, pyTorch, TensorFlow etc.)

SGD-SA (see paper)

Algorithm 2 : SGD-SA

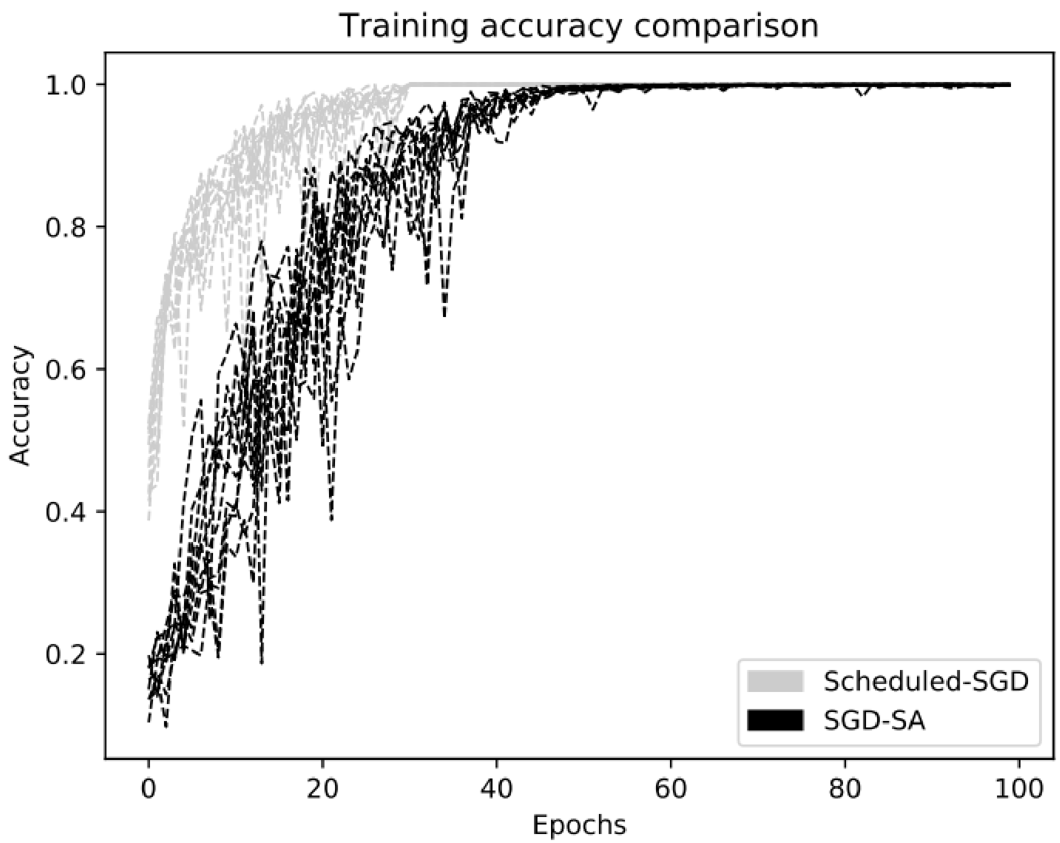
Parameters: A set of learning rates H, initial temperature $T_0 > 0$

Input: Differentiable loss function L to be minimized, cooling factor $\alpha \in (0, 1)$, number of epochs $nEpochs$, number of minibatches N

Output: the best performing $\mathrm{w}^{(i)}$ on the validation set at the end of each epoch

Divide the training dataset into N minibatches
Initialize $i = 0,\ \ T = T_0,\ \ w^{(0)} = \text{random\_initialization}()$
for $t = 1, \ldots, nEpochs$ **do**
  for $n = 1, \ldots, N$ **do**
    Extract the n-th minibatch $(x,\ y)$
    Compute $L(w^{(i)}, x, y)$ and its gradient $v = \text{backpropagation}(w^{(i)}, x, y)$
    Randomly pick a learning rate η from H
    $w_{new} = w^{(i)} - \eta\ v$
    Compute $L(w_{new}, x, y)$
    $worsening = L(w_{new}, x, y) - L(w^{(i)}, x, y)$
    $prob = e^{-worsening/T}$
    if $random(0, 1) < prob$ **then**
      $w^{(i+1)} = w_{new}$
    else
      $w^{(i+1)} = w^{(i)}$
    end if
    $i = i + 1$
  end for
  $T = \alpha \cdot T$
end for

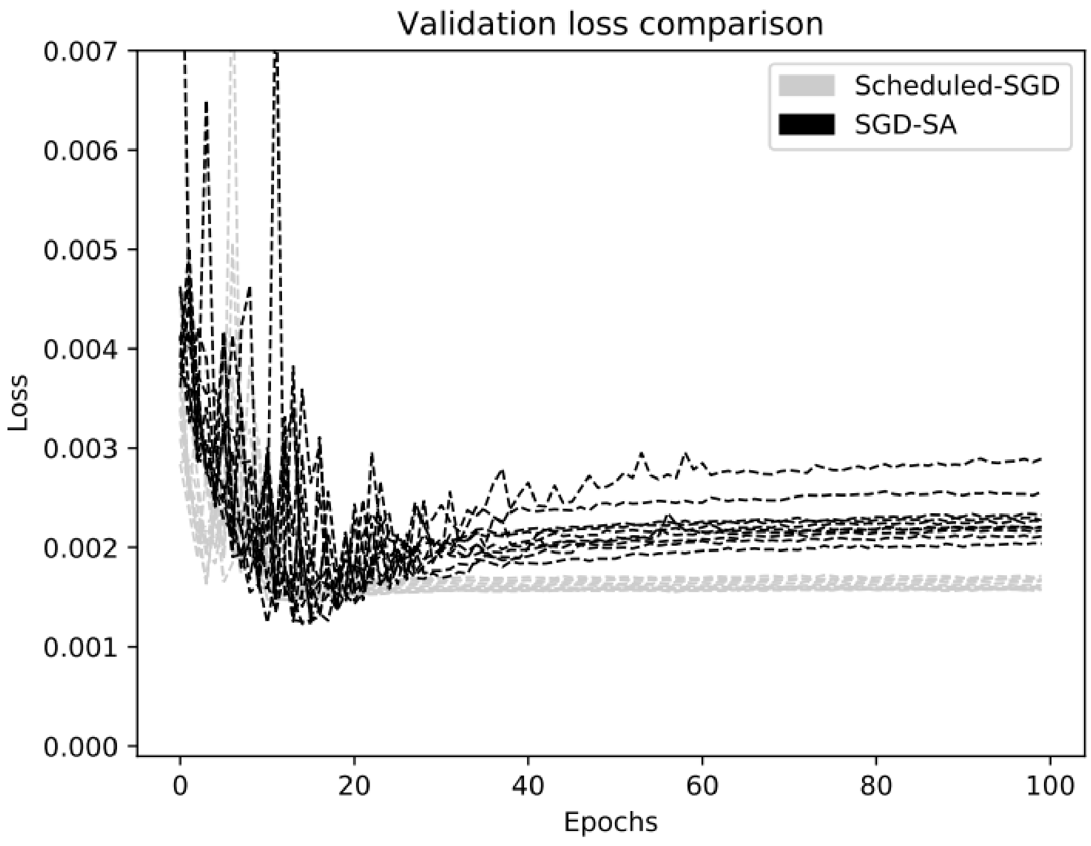
More turbulence on the training set …



(b) Training accuracy (10 runs with different random seeds)

Fig. 3: Optimization efficiency over the training set (VGG16 on CIFAR-10)

… but better results on validation!

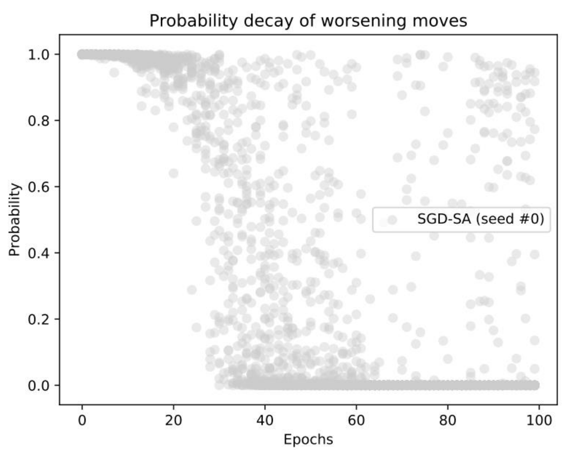


(a) Validation loss

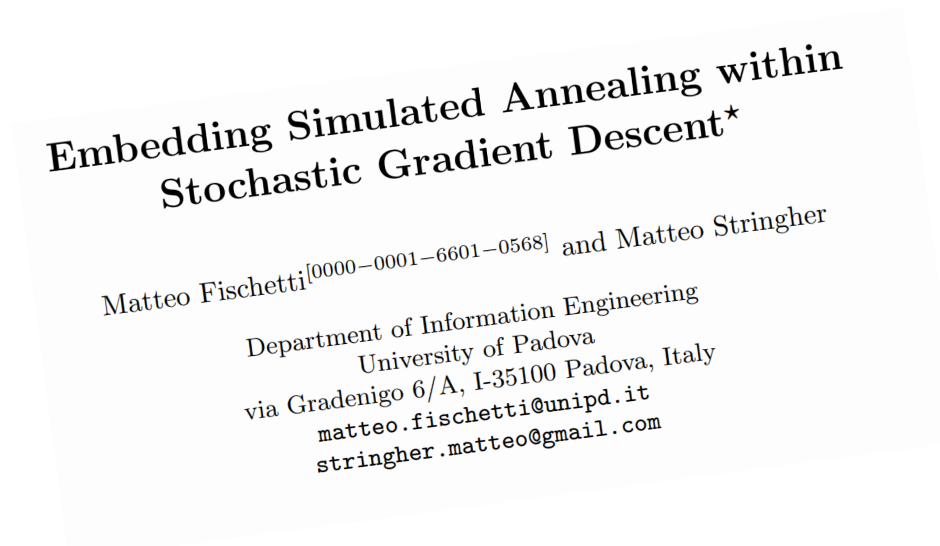
Thanks for your attention!

Slides available at
http://www.dei.unipd.it/~fisch/papers/slides/

Paper available at
http://www.dei.unipd.it/~fisch/papers/2021_embedding_SA_into_SGD.pdf



(a) Probability of accepting worsening moves



Embedding Simulated Annealing within Stochastic Gradient Descent*

Matteo Fischetti[0000−0001−6601−0568] and Matteo Stringher

Department of Information Engineering
University of Padova
via Gradenigo 6/A, I-35100 Padova, Italy
matteo.fischetti@unipd.it
stringher.matteo@gmail.com