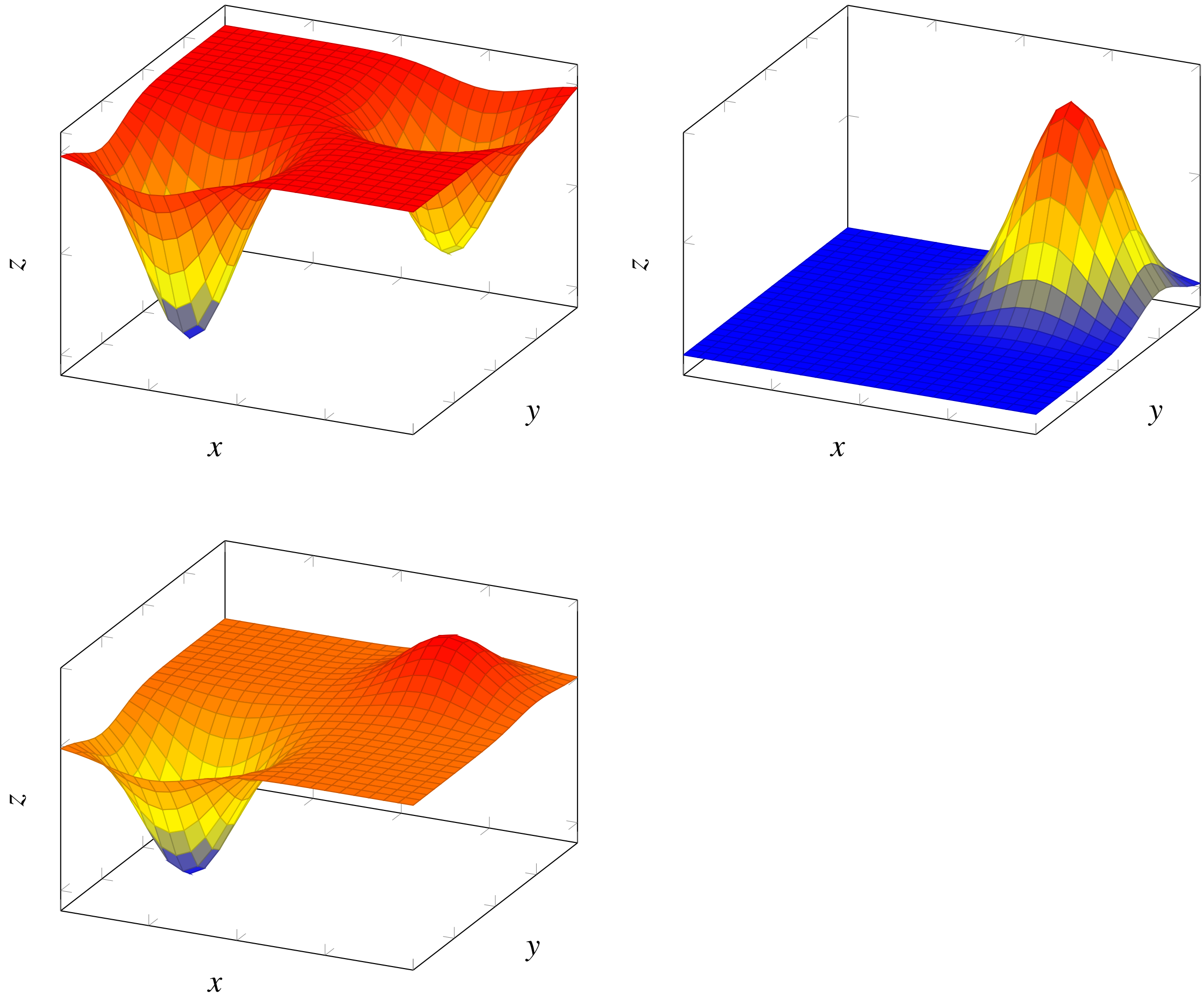



Figure 0.1: Example of a loss surface with two minima. If we place a hill at the position of the local minimum, it disappears and optimization will not get stuck.

# 1 Basics

**Algorithm 1** Stochastic gradient descent

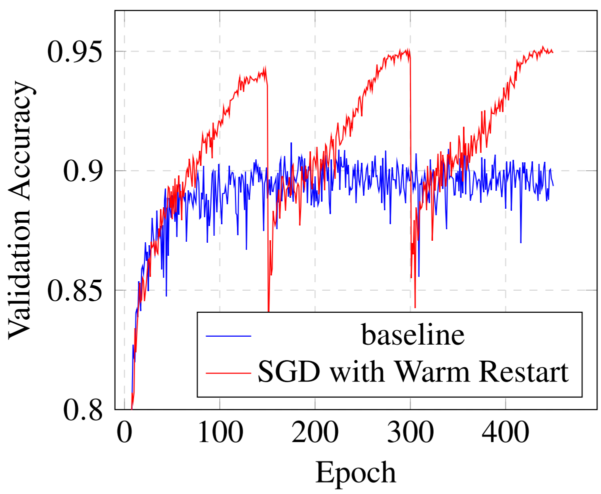
**Require:** learning rate $\lambda$
**Ensure:** a trained neural network

initialize the network, dataset and training parameters
**while** stopping criteria is not met **do**
    sample minibatch of $m$ examples $x^{(1)}, ..., x^{(m)}$
    compute gradient estimate $\hat{g} = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)})$
    apply parameter update $\theta = \theta - \lambda \cdot \hat{g}$
**end while**
**return: the trained network**

**Algorithm 2** Stochastic gradient descent with Momentum

**Require:** learning rate $\lambda$
**Require:** momentum parameter $m$
**Ensure:** a trained neural network

initialize the network, dataset and training parameters
**while** stopping criteria is not met **do**
    sample minibatch of $m$ examples $x^{(1)}, ..., x^{(m)}$
    compute gradient estimate $\hat{g} = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i);\theta}), y^{(i)})$
    compute velocity update $v = m \cdot v - \lambda \hat{g}$
    apply parameter update $\theta = \theta - v$
**end while**
**return: the trained network**



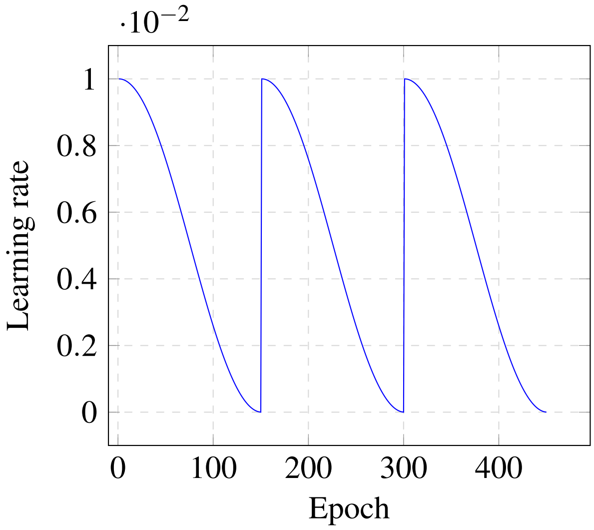




Figure 1.1: Cosine Decay with Warm Restart outperforms a fixed learning rate and increases the maximum accuracy for each restart (left). The right side shows how the learning rate is decayed.

# 2 Methods

---

**Algorithm 3** Machine Learning with distancing

---

**Require:** a set of parameters $\theta$ and a dataset
**Ensure:** a assignment of $\theta$ which maximizes performance

```
initialize the network, dataset and training parameters
for i ← 1 to desired number of epochs do
    compute foward and backward pass of training data
    update parameter values with optimizer
end for
create checkpoint we want to distance from
for i ← next epoch to end do
    for checkpoint in list of checkpoints do
        compute parameter update which maintains performance but also increases distance
        to the checkpoint
    end for
    update parameter values with optimizer
end for
return: the final assignment of θ
```

---

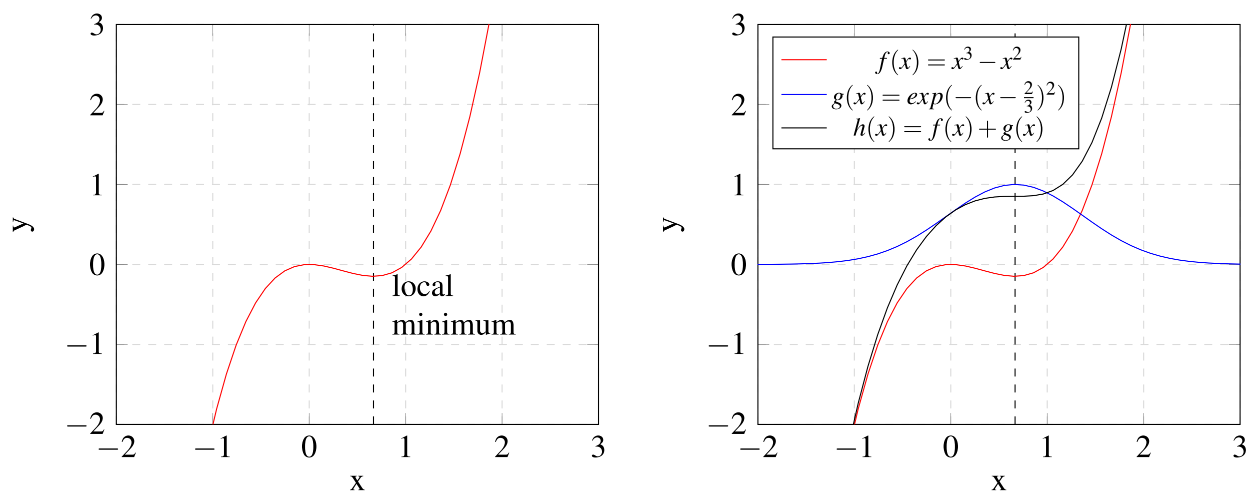



Figure 2.1: The red function has a local minimum at $x = \frac{2}{3}$. If we place a hill (blue) at this position and add the functions together the local minimum disappears (black).

**Algorithm 4** Update step with distancing

**Require:** learning rate $\lambda$, distance hyperparameters $s$ and $\sigma$

**Ensure:** a trained neural network

1: initialize the network, dataset and training parameters
2: **while** stopping criteria is not met **do**
3: sample minibatch of $m$ examples $x^{(1)}, \ldots, x^{(m)}$
4: compute gradient estimate $\hat{g} = \nabla_\theta \frac{1}{m} \sum_i L(f(x^{(i)};\theta), y^{(i)}) + s \cdot \frac{1}{c} \sum_c distance(\theta, \theta_c)$
5: apply parameter update $\theta = \theta - \lambda \cdot \hat{g}$
6: **end while**
7: **return: the trained network**

# 3 results

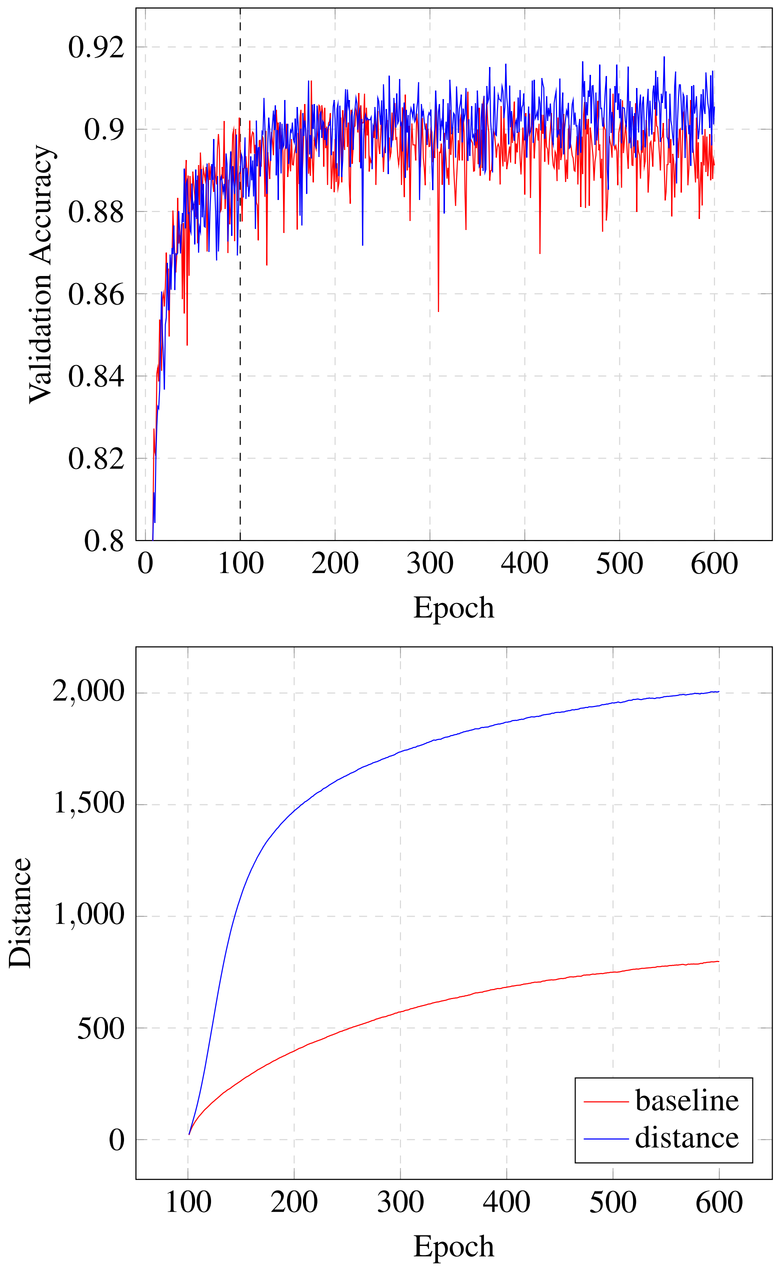
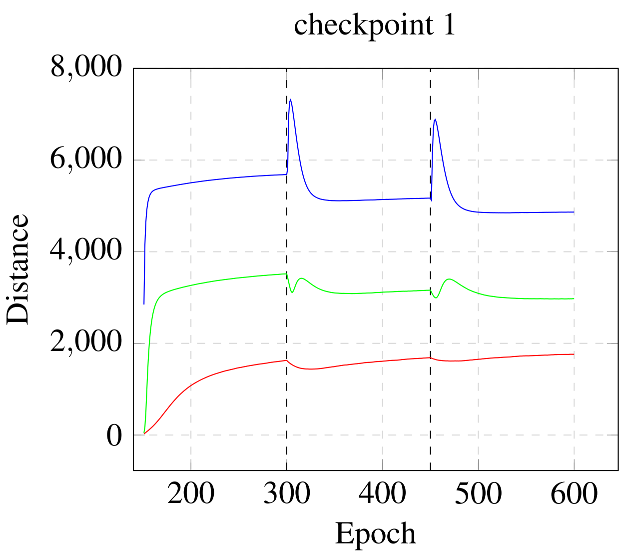



Figure 3.1: Baseline results for MobileNetV2. Upper plot shows validation accuracy, dotted line denotes addition of checkpoint. Lower plot shows $L_2$ Distance to checkpoint.



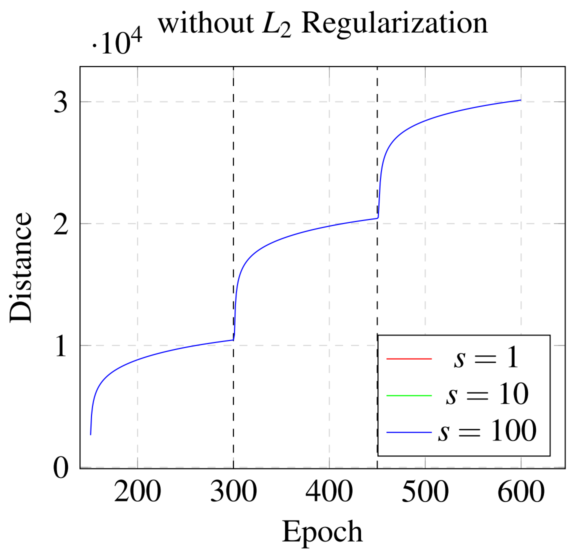




Figure 3.5: The upper plot shows the influence of new checkpoints on the distance to existing ones. Removing $L_2$ Regularization changes this behaviour (lower plot).

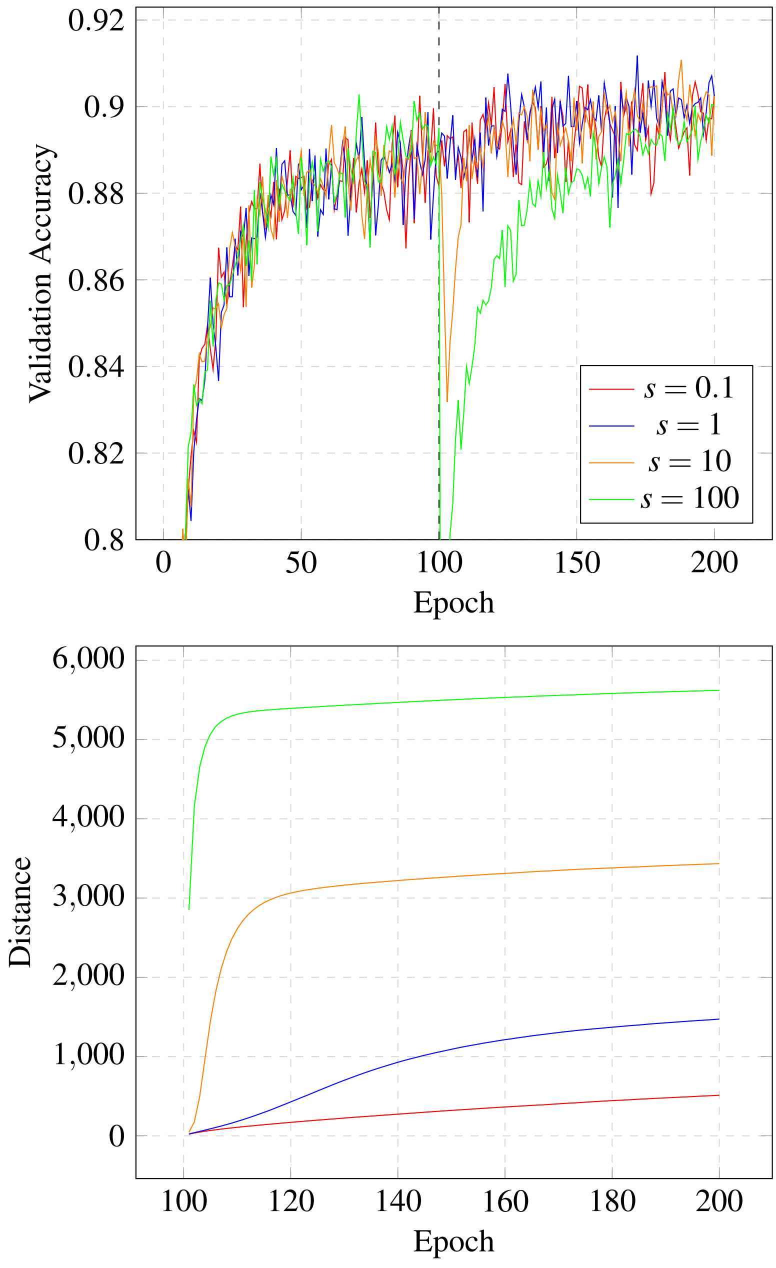



Figure 3.2: Influence of the strength hyperparameters $s$ on the validation accuracy and distance.

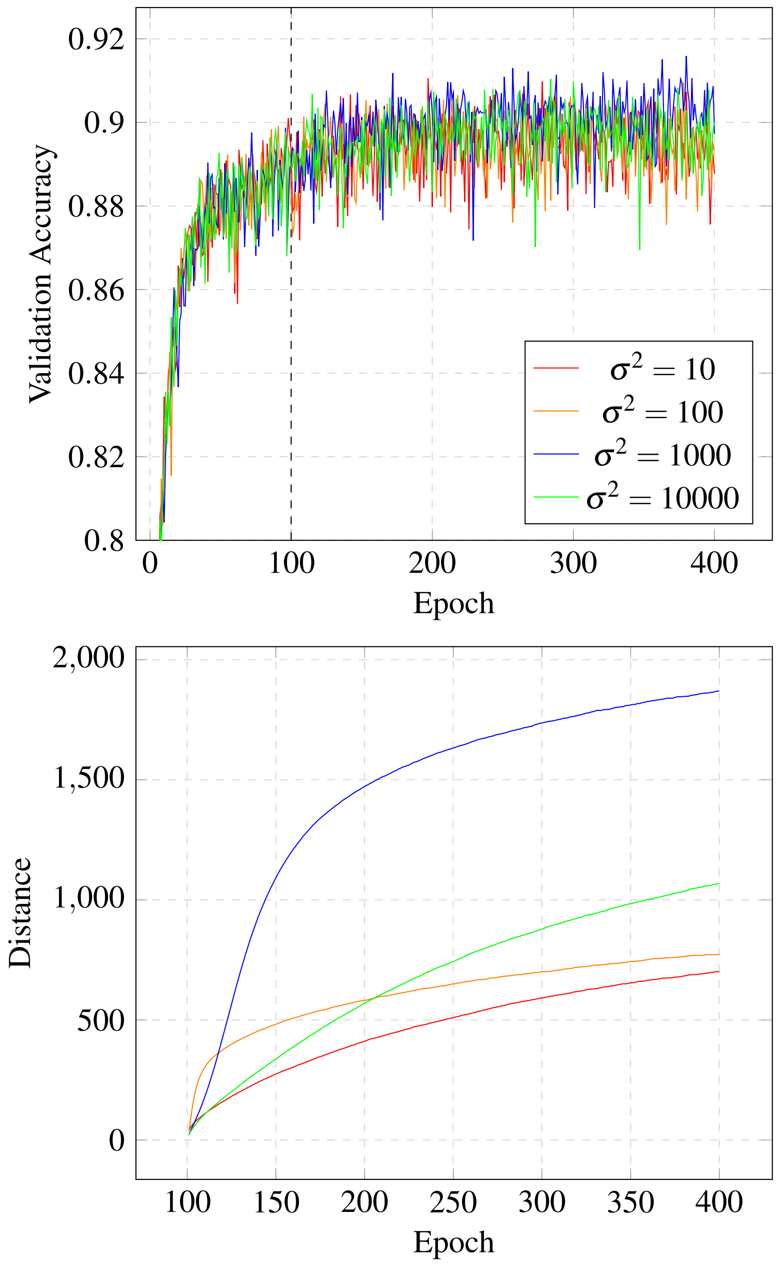



Figure 3.3: Influence of the width hyperparameters $\sigma^2$ on the validation accuracy and distance.

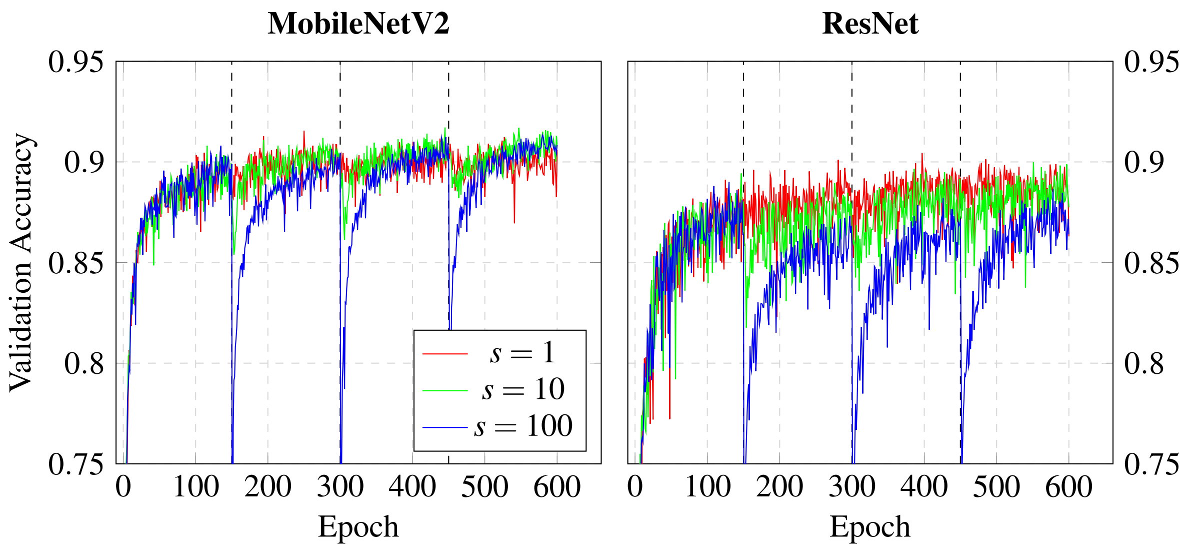



Figure 3.4: Influence of multiple checkpoints on the validation accuracy of MobileNetV2 and ResNet

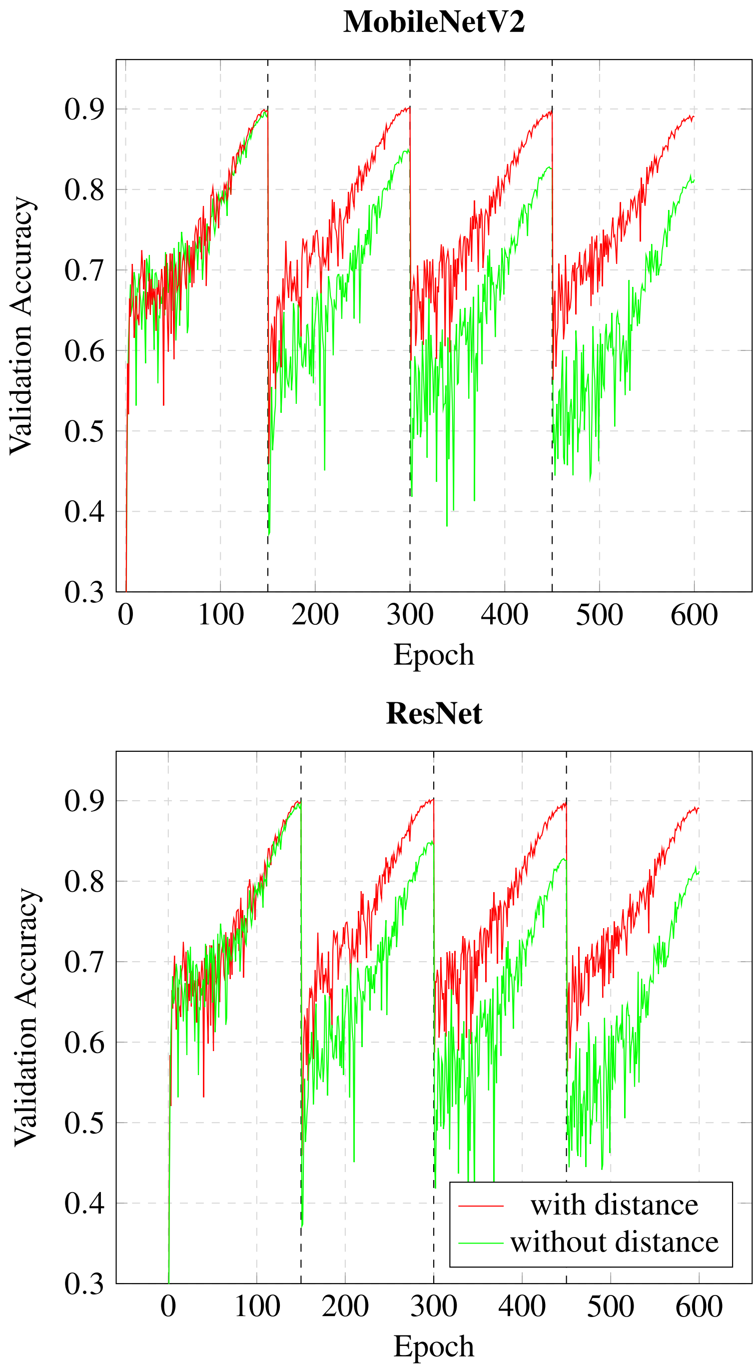



Figure 3.6: Influence of a suboptimal learning rate on Cosine Decay with Warm Restart without and with distance function.

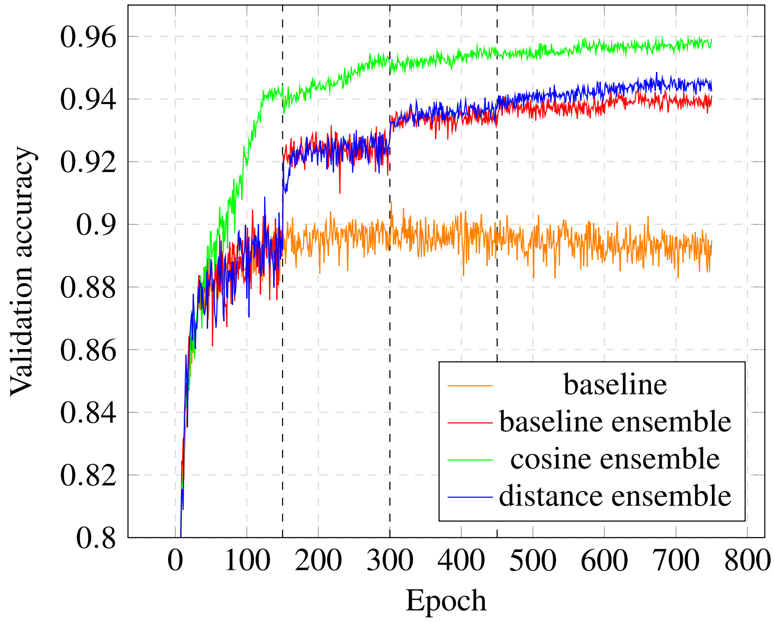



Figure 3.7: Ensemble accuracy for different networks against the baseline accuracy of a single network.

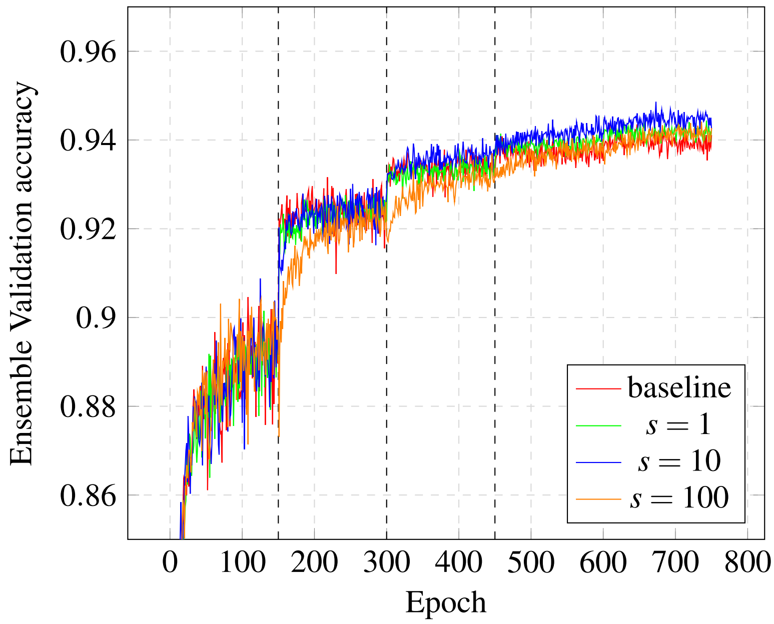
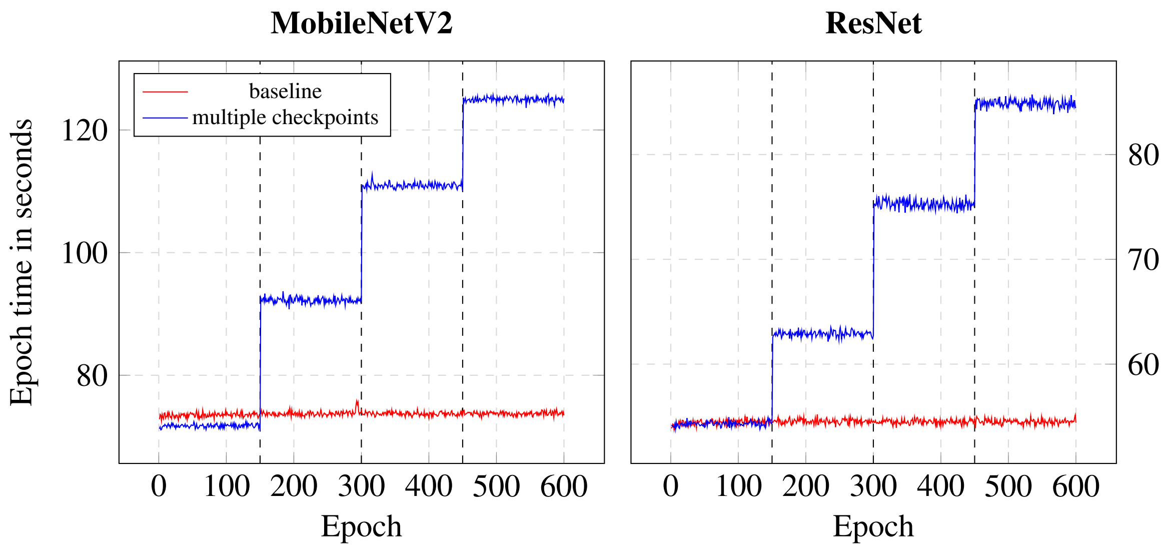



Figure 3.8: Ensemble accuracy for different strength values.



MobileNetV2
ResNet
baseline
multiple checkpoints
Epoch time in seconds
120
100
80
70
60
0
100
200
300
400
500
600
Epoch
Epoch