0.1 Basics

Algorithm 1 Stochastic gradient descent

Require: learning rate λ
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 $x^{(1)},...,x^{(n)}$
4: compute gradient estimate $\hat{g} = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i)};\theta),y^{(i)})$
5: apply parameter update $\theta = \theta - \lambda \cdot \hat{g}$
6: **end while**
7: **return: the trained network**

Algorithm 2 Stochastic gradient descent with Momentum

Require: learning rate α
Require: momentum parameter m
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 $x^{(1)},...,x^{(n)}$
4: compute gradient estimate $\hat{g} = \frac{1}{m}\nabla_\theta \sum_i L(f(x^{(i);\theta}),y^{(i)})$
5: compute velocity update $v = m \cdot v - \alpha\hat{g}$
6: apply parameter update $\theta = \theta - v$
7: **end while**
8: **return: the trained network**

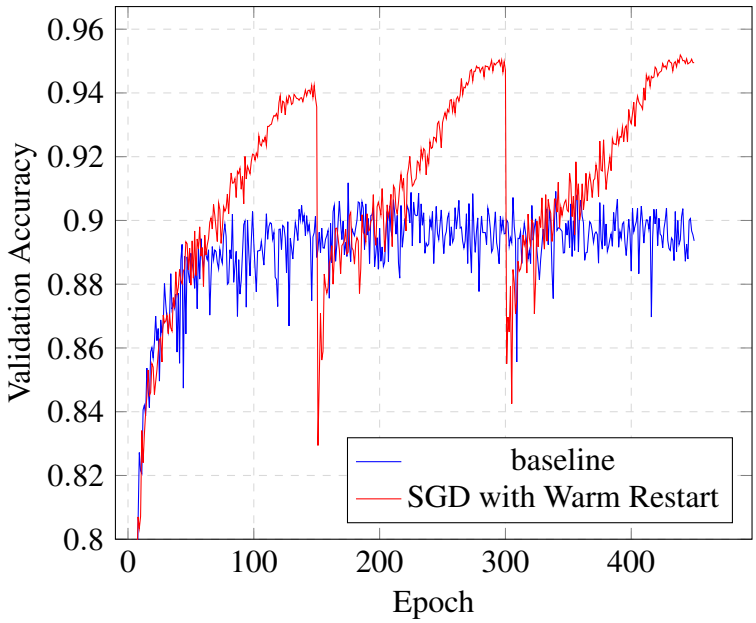
0.2 Methods

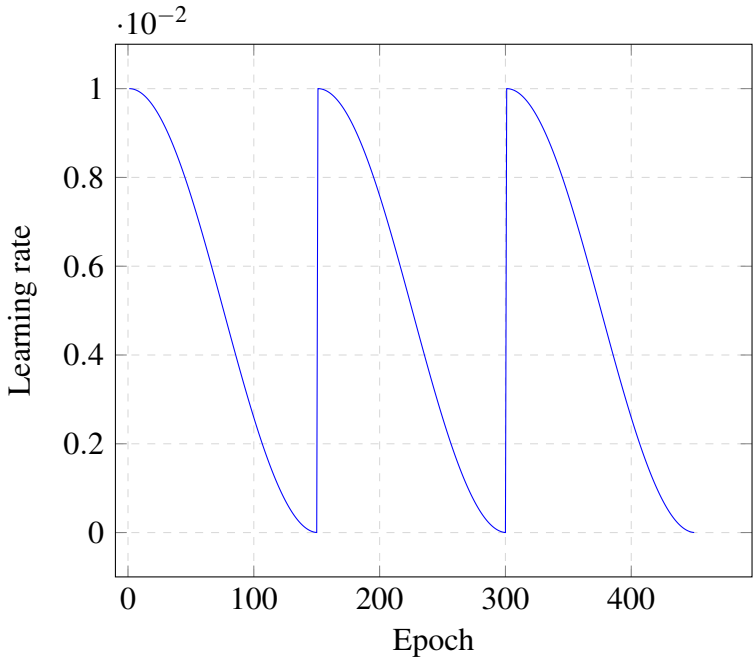
Algorithm 3 Machine Learning with distancing

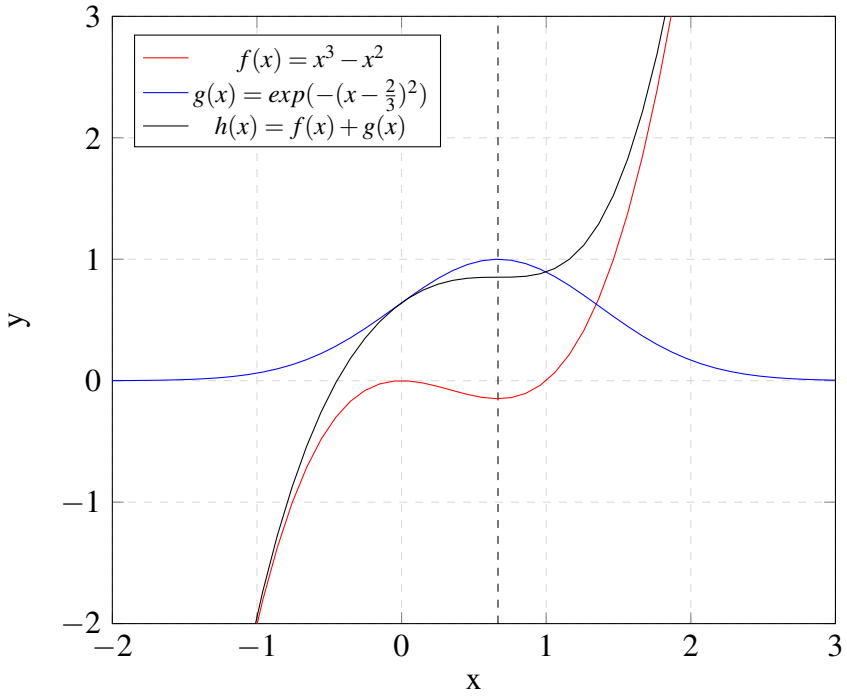
Require: a set of parameters θ and a dataset
Ensure: a assignment of θ which maximizes performance
initialize the network, dataset and training parameters
for $i \leftarrow 1$ **to** desired number of epochs **do**
    compute foward and backward pass of training data
    update parameter values with optimizer
end for
create checkpoint we want to distance from
for $i \leftarrow$ next epoch **to** end **do**
    for checkpoint **in** list of checkpoints **do**
        compute parameter update which increases the distance to checkpoint
    end for
    update parameter values with optimizer
end for
return: the final assignment of θ

Algorithm 4 Update step with distancing

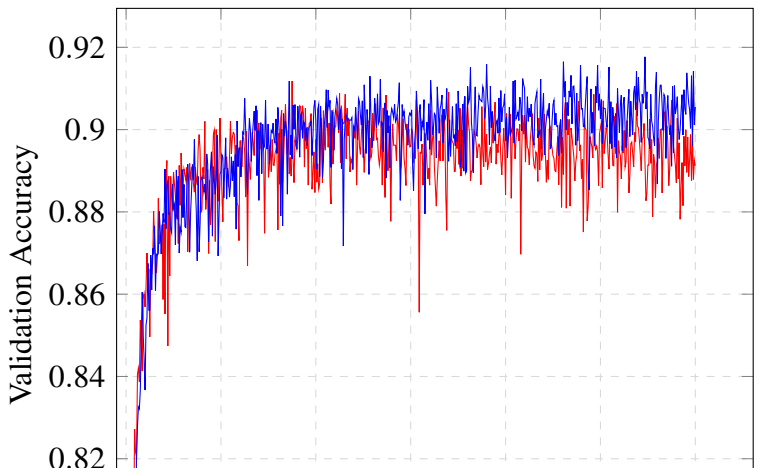
Require: learning rate α, distance hyperparameters s and σ
Ensure: a trained neural network
initialize the network, dataset and training parameters
while stopping criteria is not met **do**
    sample minibatch of m examples $x^{(1)}, ..., x^{(n)}$
    compute gradient estimate $\hat{g} = \nabla_\theta \frac{1}{m} \sum_i L(f(x^{(i)}; \theta), y^{(i)}) + \frac{1}{c} \sum_c s_c \cdot distance(\theta, \theta_c)$
    apply parameter update $\theta = \theta - \alpha \cdot \hat{g}$
end while
return: the trained network

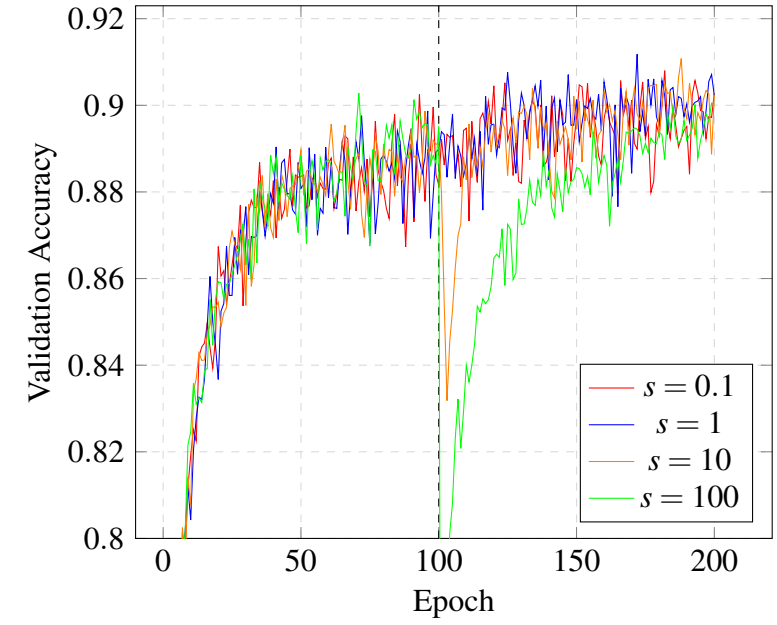

0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
Validation Accuracy
0
100
200
300
400
Epoch
baseline
SGD with Warm Restart

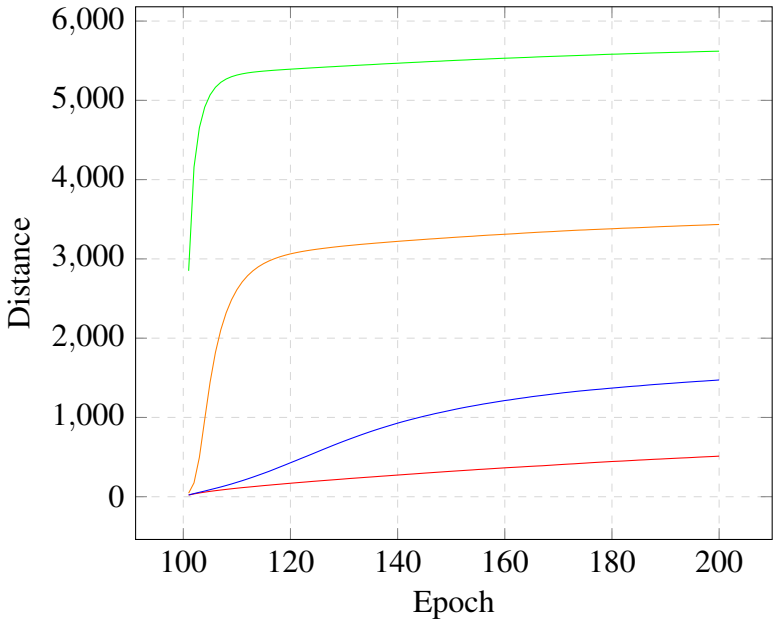

$\cdot 10^{-2}$
Learning rate
Epoch
0
0.2
0.4
0.6
0.8
1
0
100
200
300
400

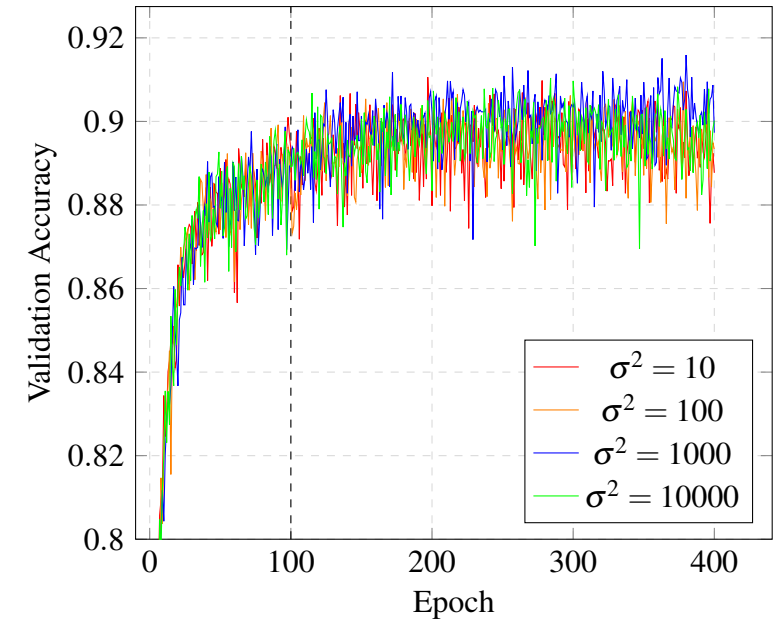

$f(x) = x^3 - x^2$
$g(x) = exp(-(x - \frac{2}{3})^2)$
$h(x) = f(x) + g(x)$
y
x
3
2
1
0
−1
−2
−2
−1
0
1
2
3

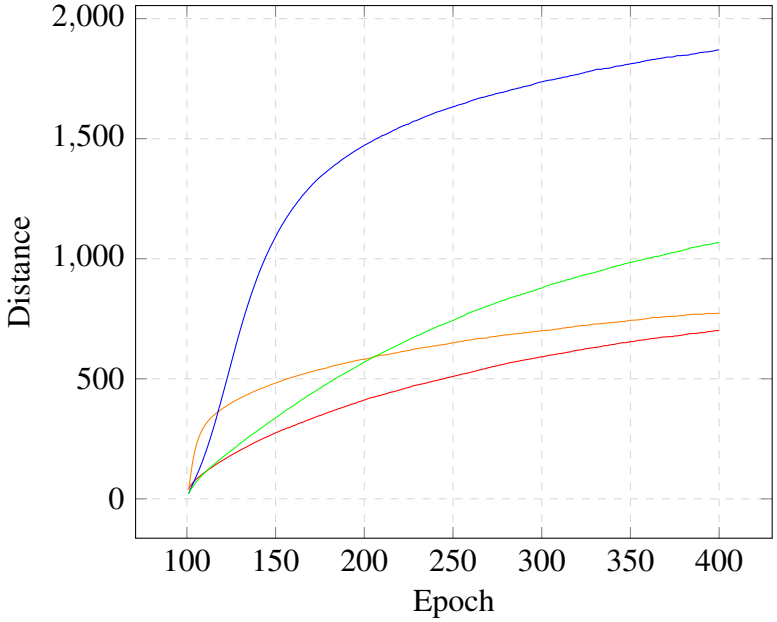
0.3 results


Validation Accuracy
0.92
0.9
0.88
0.86
0.84
0.82

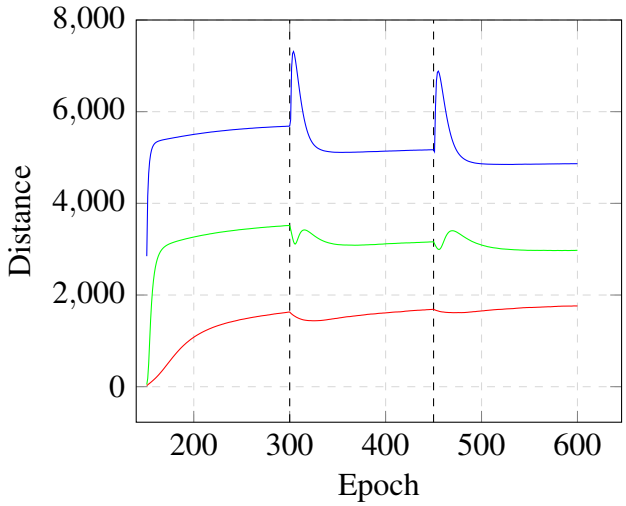

0.92
0.9
0.88
0.86
0.84
0.82
0.8
Validation Accuracy
0
50
100
150
200
Epoch
$s = 0.1$
$s = 1$
$s = 10$
$s = 100$


6,000
5,000
4,000
3,000
2,000
1,000
0
Distance
100
120
140
160
180
200
Epoch

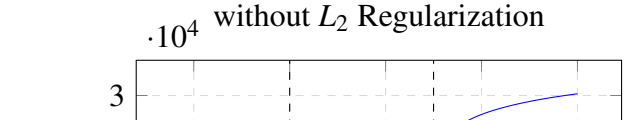


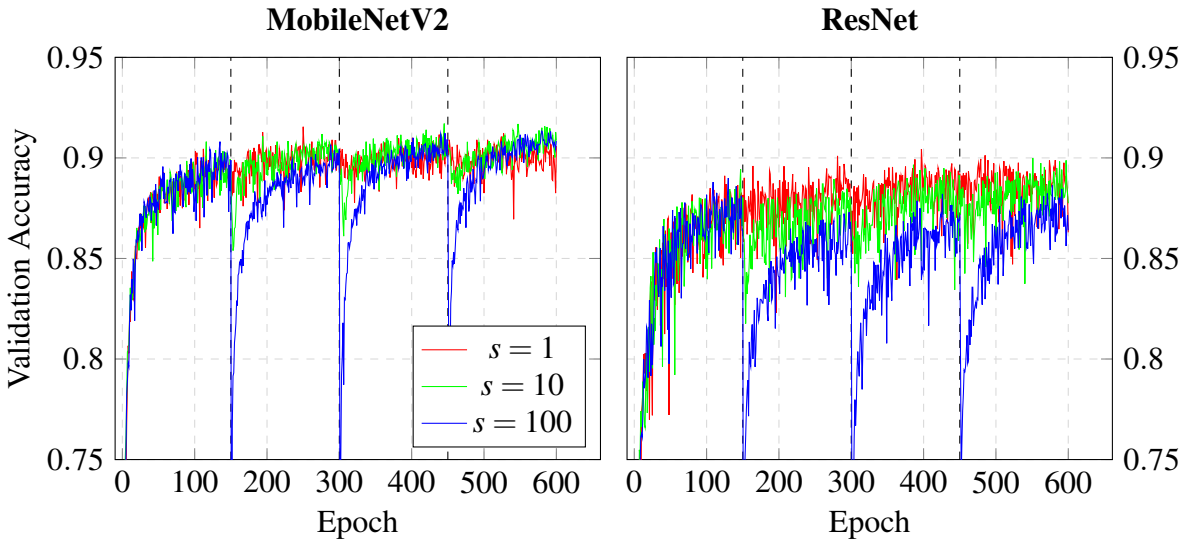


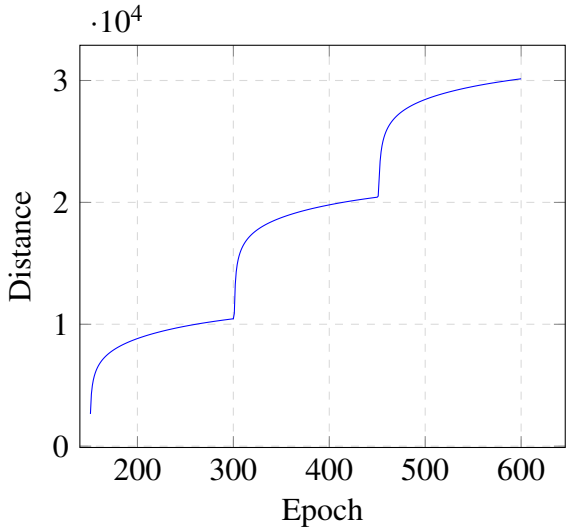
checkpoint 1

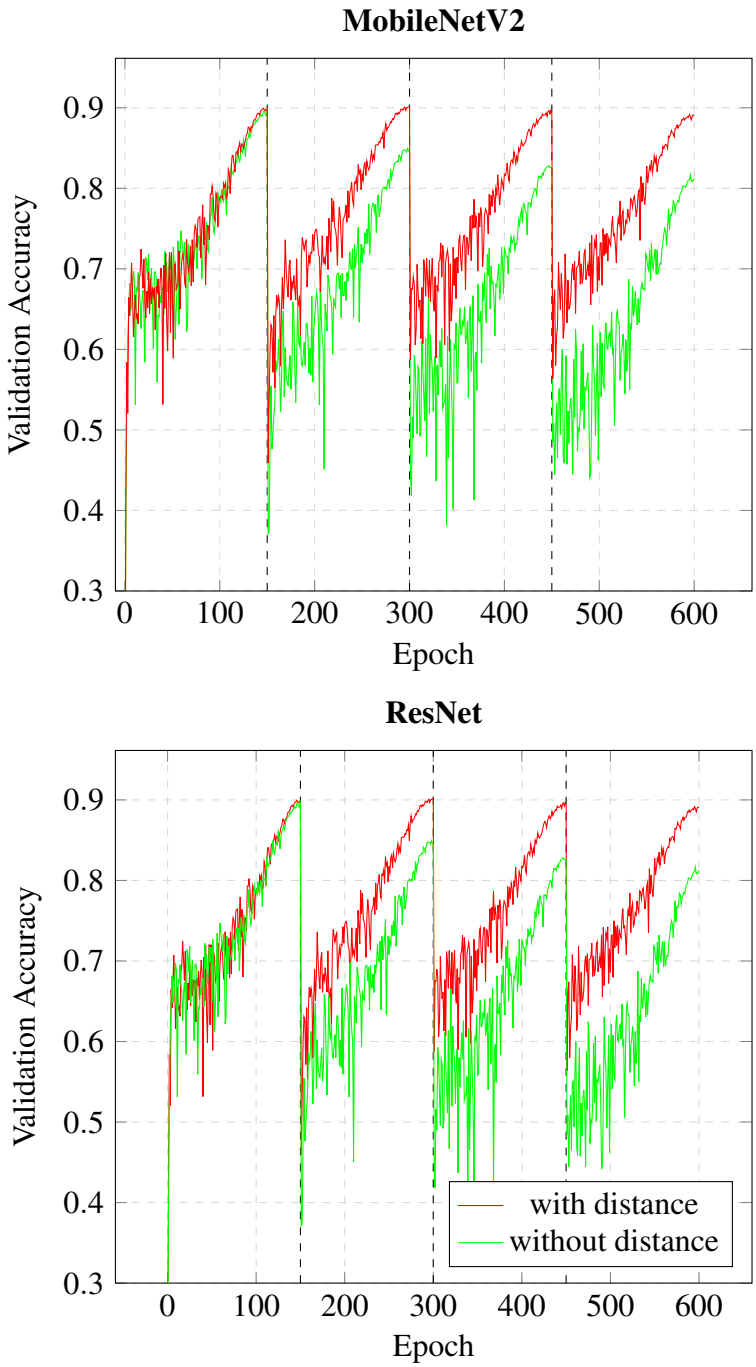


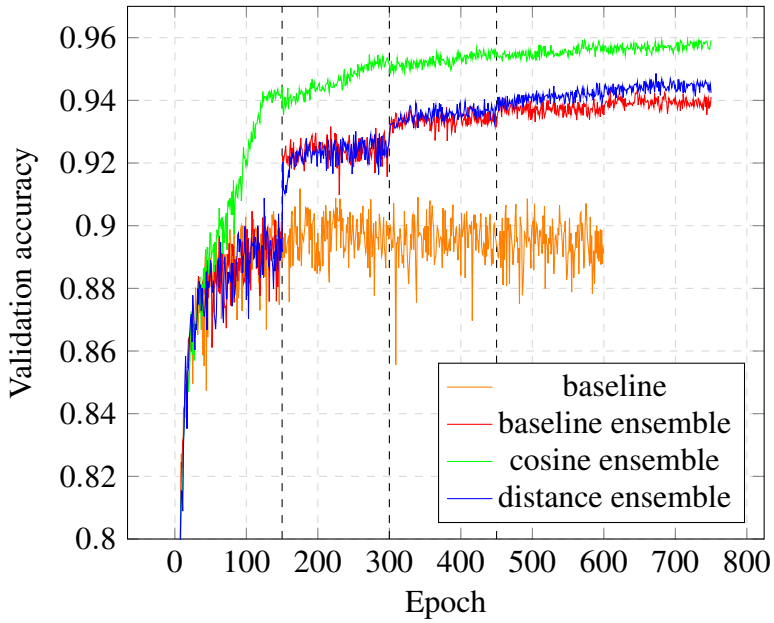
without L_2 Regularization

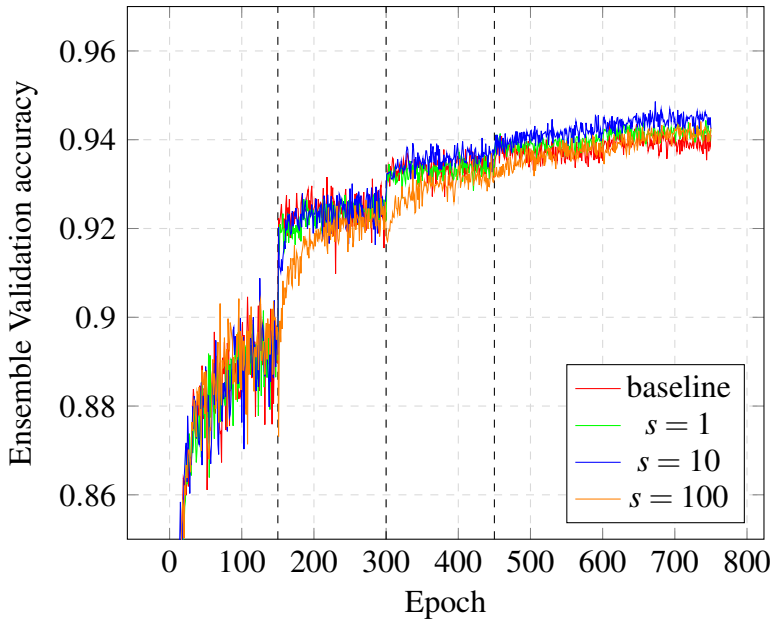


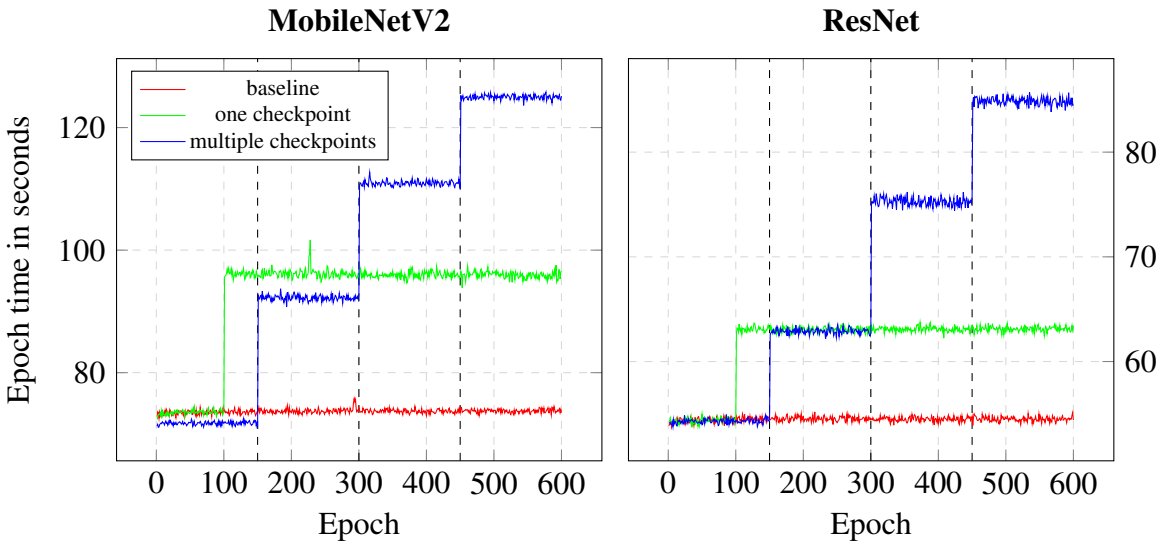

MobileNetV2
ResNet
Validation Accuracy
0.95
0.9
0.85
0.8
0.75
0
100
200
300
400
500
600
Epoch
$s = 1$
$s = 10$
$s = 100$


$\cdot 10^4$
Distance
3
2
1
0
200
300
400
500
600
Epoch


MobileNetV2
Validation Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
100
200
300
400
500
600
Epoch
ResNet
Validation Accuracy
0.9
0.8
0.7
0.6
0.5
0.4
0.3
with distance
without distance
0
100
200
300
400
500
600
Epoch


Validation accuracy
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0
100
200
300
400
500
600
700
800
Epoch
baseline
baseline ensemble
cosine ensemble
distance ensemble


Ensemble Validation accuracy
0.96
0.94
0.92
0.9
0.88
0.86
0
100
200
300
400
500
600
700
800
Epoch
baseline
$s = 1$
$s = 10$
$s = 100$


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