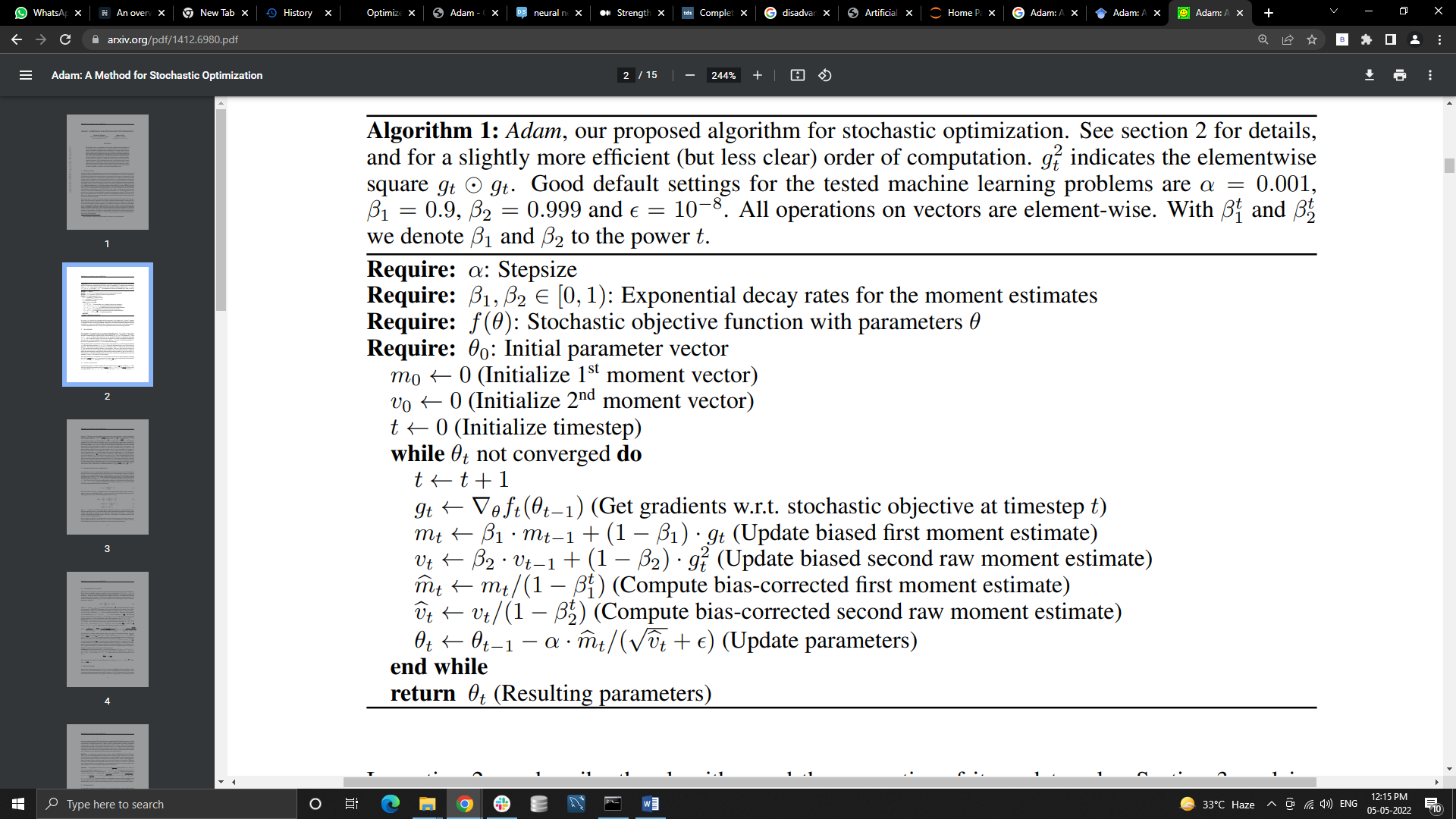
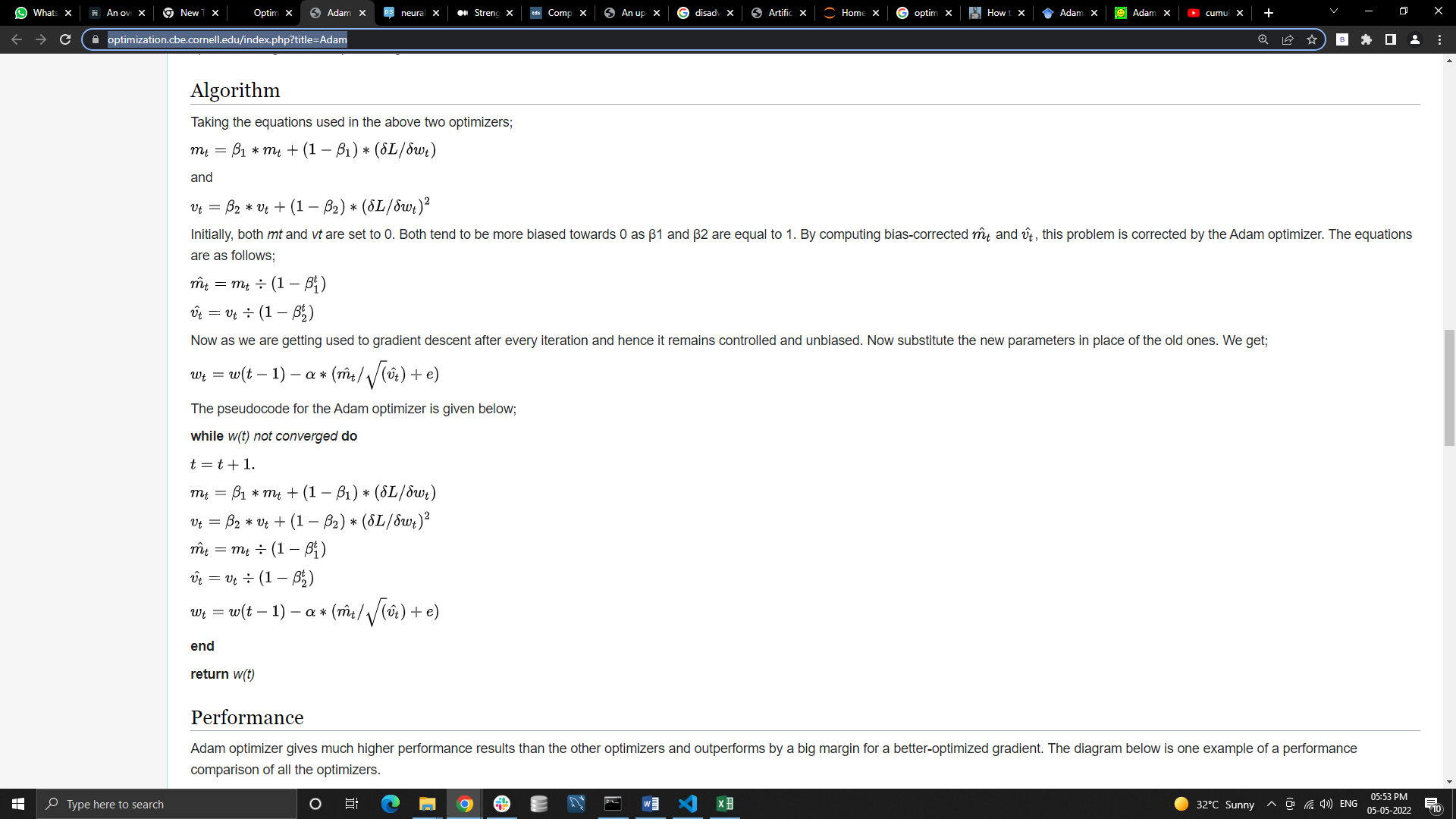
What is Adam Optimizer (Paper: <https://arxiv.org/abs/1412.6980> | <https://doi.org/10.48550/arXiv.1412.6980> | <https://arxiv.org/pdf/1412.6980.pdf> )

663

<https://optimization.cbe.cornell.edu/index.php?title=Adam>



The Adam algorithm is a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement.

This algorithm used to optimize an objective function f(θ), with parameters θ (weights and biases).

Implementation of Adam using TensorFlow, PyTorch, and NumPy

TensorFlow :

import tensorflow as tf

tf.keras.optimizers.Adam(

learning\_rate=0.001,

beta\_1=0.9,

beta\_2=0.999,

epsilon=1e-07,

amsgrad=False,

name='Adam')

<https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam>

PyTorch :

import torch

torch.optim.Adam(

params,

lr=0.001,

betas=(0.9, 0.999),

eps=1e-08,

weight\_decay=0,

amsgrad=False)

<https://pytorch.org/docs/stable/optim.html>

NumPy :

<https://gist.github.com/LayanCS/8bc7b0ec110487dd1e485c809954b044>

<https://github.com/LayanCS>

Advantages and Disadvantages of Adam:

Advantages:

1. Can handle sparse gradients on noisy datasets.
2. Default hyper parameter values do well on most problems.
3. Computationally efficient.
4. Requires little memory, thus memory efficient.
5. Works well on large datasets.

Disadvantages:

1. Adam does not converge to an optimal solution in some areas (this is the motivation for AMSGrad).
2. Adam can suffer a weight decay problem (which is addressed in AdamW).
3. Recent optimization algorithms have been proven faster and better (<https://johnchenresearch.github.io/demon/> ).

Adam is also an adaptive gradient descent algorithm and it optimize a learning rate, weights and biases per-parameter function f(θ). But Adam not able to optimize in some areas and also suffer a weight decay problem.

Practitioners noticed that in some cases, like object recognition or machine translation Adam fail to give an optimal solution. The exponential moving average of past squared gradients as a reason for the poor generalization behaviour of adaptive learning rate methods. To resolve this issue the proposed algorithm keep running deflection and average on mt and vt . If the deflection is greater than the threshold then the average value is used to smoothing of weight.

X1 , x2

X2-x1

((X2-x1)/ x1) \* 100