WebNN: A Distributed Framework for Deep Learning

Aaron Goin, Ronald Cotton and Xinghui Zhao
School of Engineering and Computer Science
Washington State University, Vancouver WA 98686, U.S.A.

Introduction

- Distributed computing technologies have been leveraged in machine learning applications
- High-end resources are used, e.g. cluters, GPUs etc.
- We develop a framework for serving and training deep neural networks over the internet.

Background

Tensorflow

- The most widely used framework for deep learning
- Recently, TensorFlow.js is added to the framework which brings machine learning to the browswer
- However, it requires additional services on the backend to distribute trained models to the clents

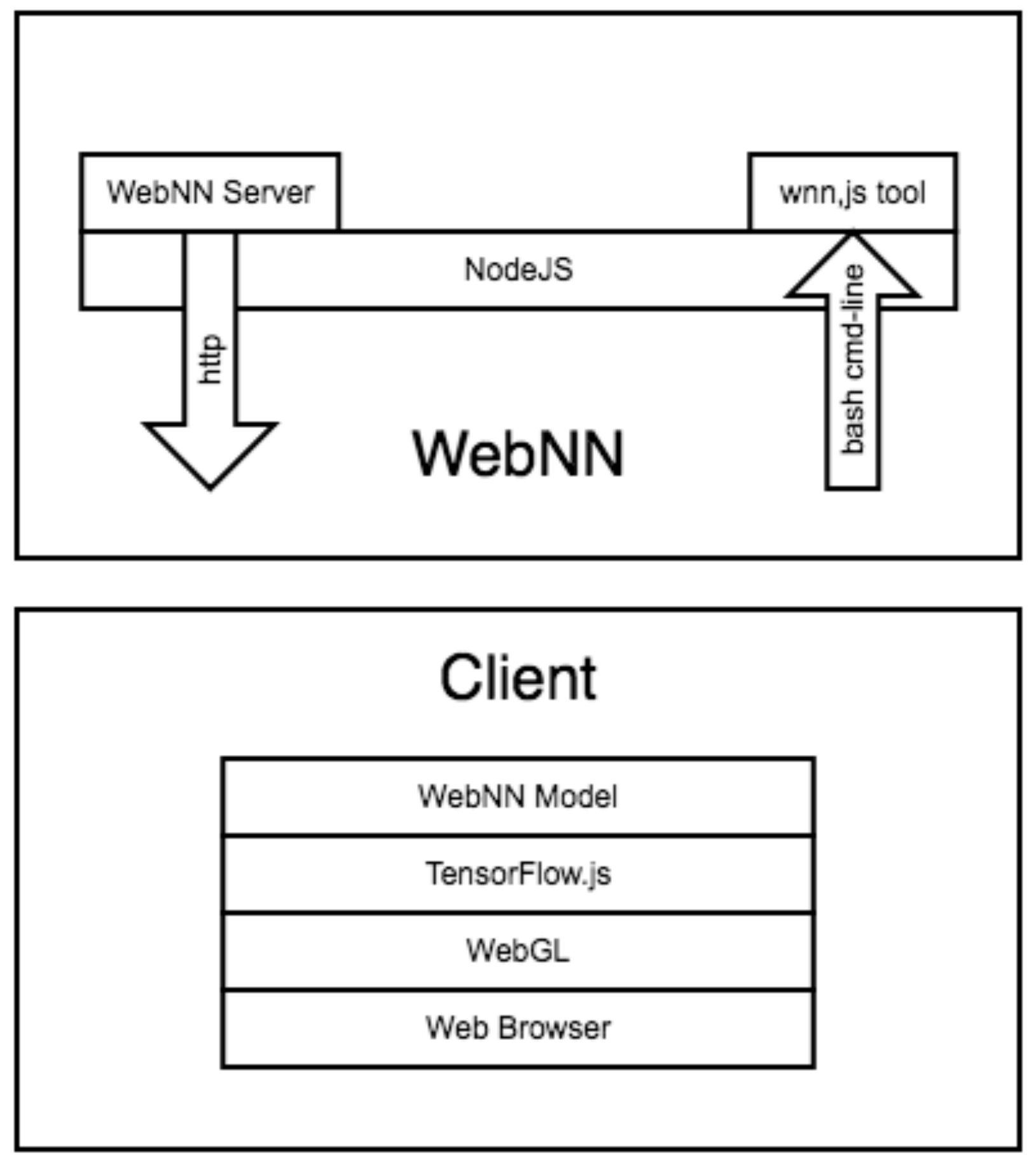
Objectives

- Develop a framework to facilitate distributing and training models in the web browser
- Effectively merge weights generated by a number of clients
- Efficiently utilize peer-owned resources distributed over the Internet

WebNN Architecture

System Design and Implementation

- WebNN server builds off of NodeJS to serve over http
- Command-line controls are supported by wnn.js
- WebNN can be deployed as a standalone server or applied to an existing application as a service



Hardware

- Both server and clients are iMacs
- 2.5GHz i5 CPU, 8GB RAM
- AMD Radeon 6750M GPU

Experimental Results

- Methodology for merging weights from multiple clients
  - Average merge: weights are simply averaged together

$$W_n = (W_{n-1} + W_p)/2.$$

  - Weighed merge: takes potential staleness into account while averaging the weights

$$W_n = (W_{n-1} * t_{n-1} + W_p * t_p)/(t_{n-1} + t_p)$$

  - Mimic merge: uses the same information as weighted average, but handles differently:

$$d = t_{n-1} - t_p$$

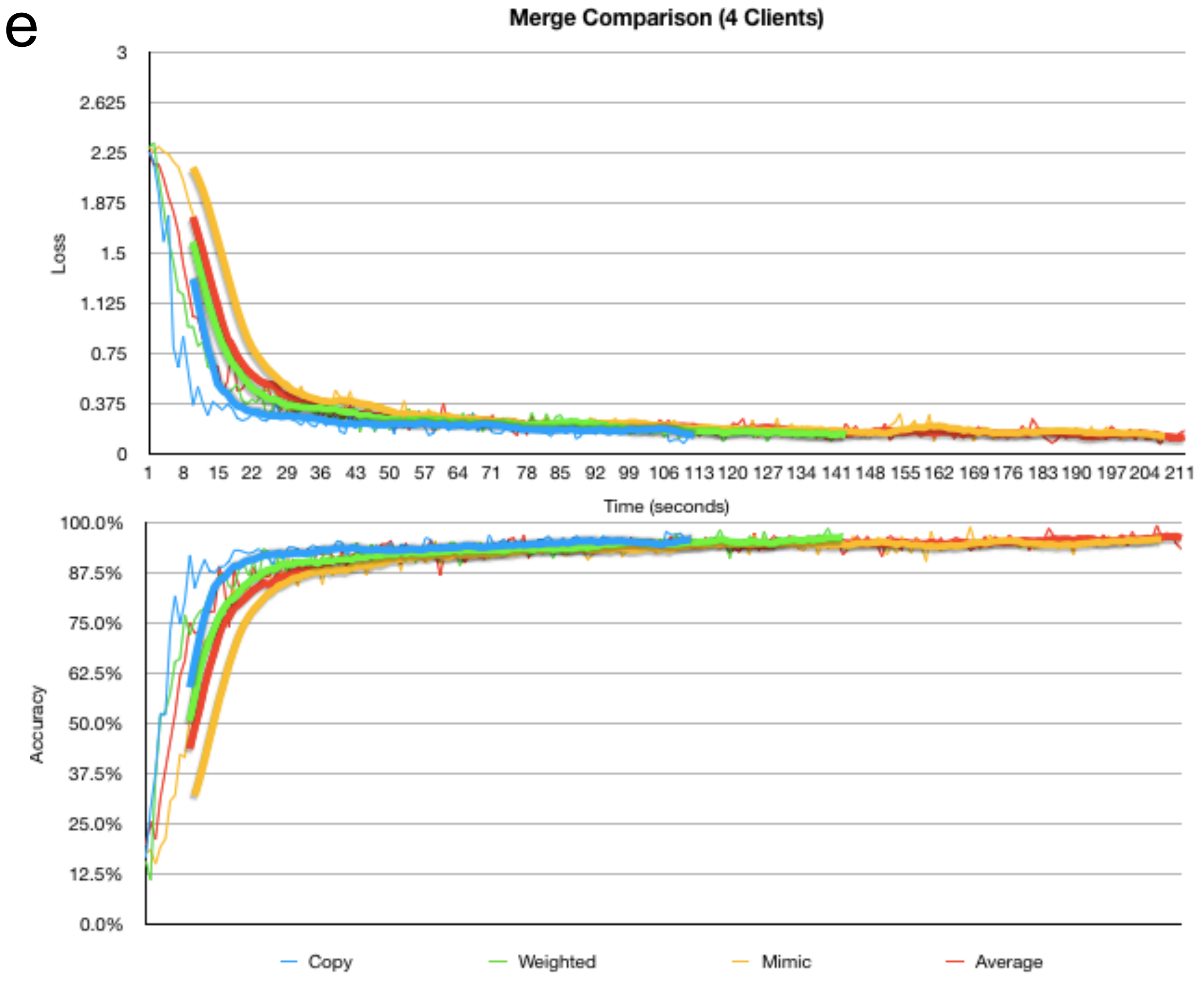
if d > 0:

$$W_D = W_p - W_{n-1}$$

else:

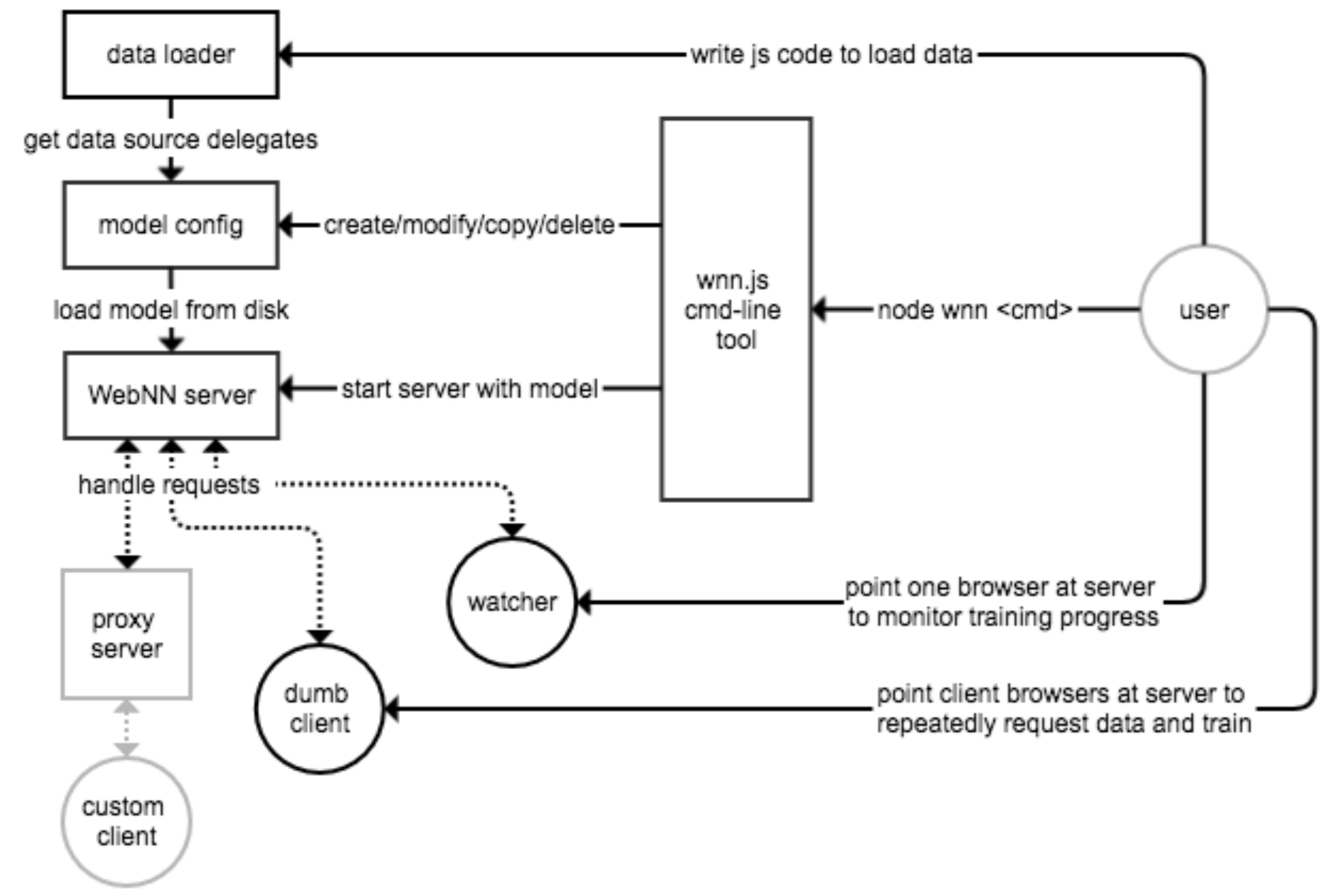
$$W_D = W_{n-1} - W_p$$

$$W_n = (W_{n-1} + W_D)/(|d| + 1).$$



Workflow

- Users create their models in a JSON format
- Configure its training and validation properties
- Create a JavaScript module used by the server to get training and validation data
- Server hands the model off to clients for training, along with a set of weights and training data upon request
- Clients send back their modified weights to the server, and receive a new set of weights to merge into their own.


data loader
write js code to load data
get data source delegates
model config
create/modify/copy/delete
wnn.js cmd-line tool
load model from disk
node wnn <cmd>
user
WebNN server
start server with model
handle requests
watcher
point one browser at server to monitor training progress
proxy server
dumb client
point client browsers at server to repeatedly request data and train
custom client

Conclusions

- WebNN is a framework for distributing and training a centralized neural network in the browser.
- WebNN can be easily deployed over a network of loosely coupled computational resources
- A peer-based weight merge system works best with a weighted average favoring weights with more training iterations behind them
- The peer-based merging can be improved to promote less variance between clients.

Acknowledgement

Special thanks to the support from WSUV Mini Research Grant.

https://labs.wsu.edu/dsr/