

ECE-471 Selected Topics in Machine Learning
Prof. Curro
Midterm Project

Evan Bubniak, Zhang Jinhan

October 27, 2019

1 Results

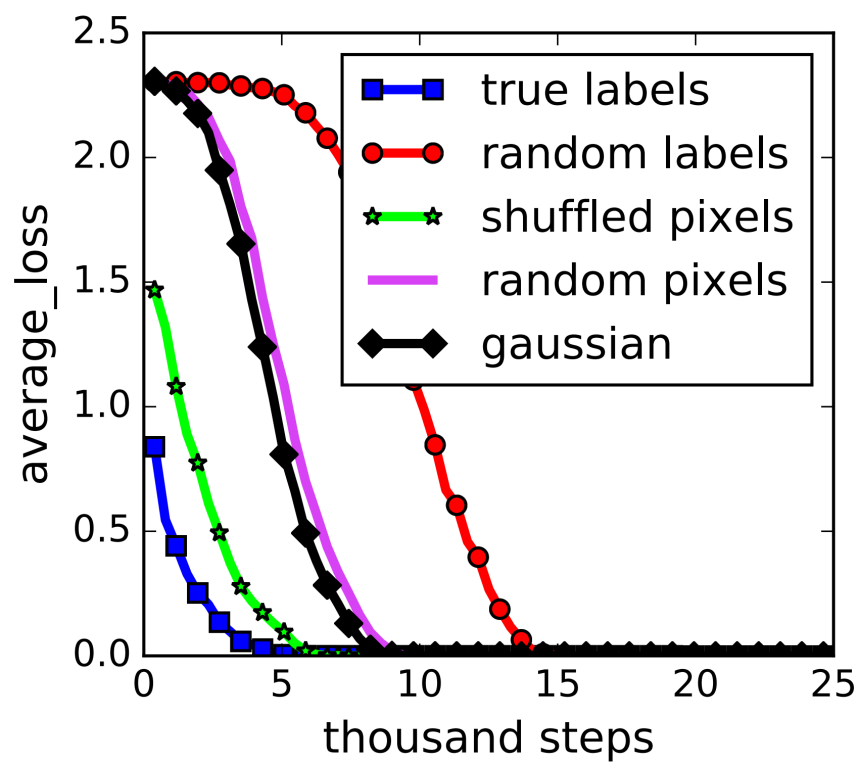


Figure 1: Juxtaposition of the noisy data points, the noiseless sinewave they are based on, and the manifold of the stochastic gradient descent regression model.

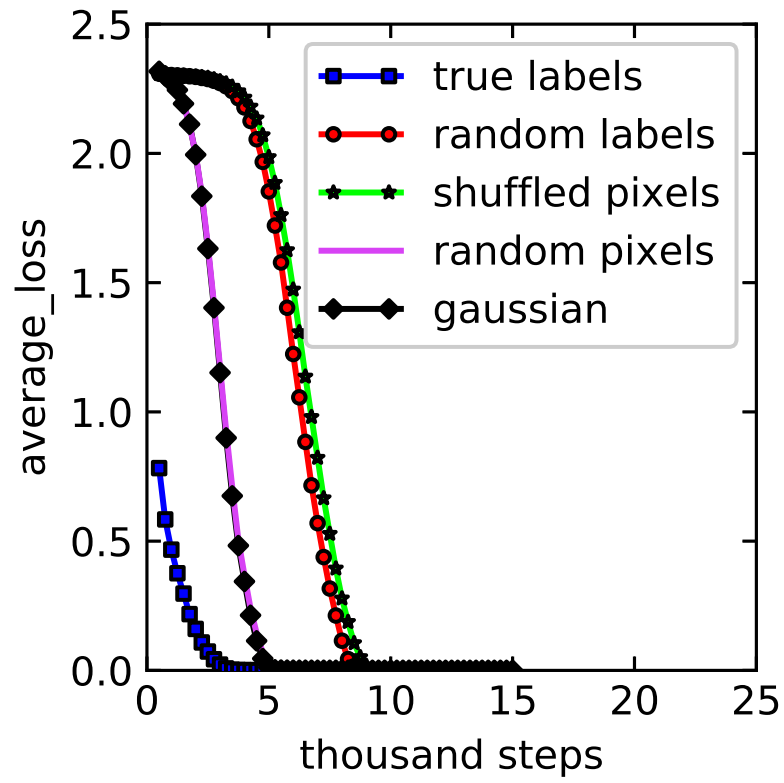


Figure 2: A plot of each of the basis functions, with the weights and intercept removed.

2 Code

```
import tensorflow.keras as keras
from tensorflow.keras.datasets import cifar10
from utils import *
from math import ceil
import sys

BATCH_SIZE = 200
NUM_SAMPLES = 50000
STEPS_PER_EPOCH = ceil(NUM_SAMPLES / BATCH_SIZE)

def get_model(model_code):
    if model_code == 1:
        return MiniInceptionV3(
            input_shape = X.shape[1:],
            num_labels = 10
        )
    elif model_code == 2:
        return MiniInceptionV3(
            input_shape = X.shape[1:],
            num_labels = 10,
            use_batch_norm = False
        )
    elif model_code == 3:
        return AlexNet(
            input_shape = X.shape[1:],
            num_labels = 10
        )
    elif model_code == 4:
        return MLP(
            input_shape = X.shape[1:],
            num_labels = 10,
            num_hidden_layers = 1
        )
    elif model_code == 5:
        return MLP(
            input_shape = X.shape[1:],
```

```

        num_labels = 10,
        num_hidden_layers = 3
    )

model_codes = [1, 2, 3, 4, 5]

(X_train, y_train), (X_test, y_test) = cifar10.load_data()

X_train = preprocess_input(X_train)
X_test = preprocess_input(X_test)
y_train = preprocess_labels(y_train)
y_test = preprocess_labels(y_test)

CORRUPTION_TYPE = ["true_labels", "random_labels", "shuffled_pixels", "random_pixels"]
NUM_STEPS = [10000, 10000, 10000, 10000, 15000]
true_inputs = [X_train, y_train]
random_labels = [X_train, randomize_labels(y_train.shape[0], 10)]
shuffled_pixels = [shuffle_pixels(X_train), y_train]
random_pixels = [randomize_pixels(X_train), y_train]
gaussian = [create_gaussian_noise(X_train), y_train]
DATA_INPUTS = [true_inputs, random_labels, shuffled_pixels, random_pixels, gaussian]

for model_code in model_codes:
    for job_name, step_count, data_input in zip(CORRUPTION_TYPE, NUM_STEPS, DATA_INPUTS):
        X = data_input[0]
        y = data_input[1]
        num_epochs = ceil(step_count / STEPS_PER_EPOCH)
        model = get_model(model_code)
        model.compile()
        model.fit(*data_input, X_test, y_test, num_epochs, job_name, BATCH_SIZE)
        model.evaluate(X_test, y_test)

```
