Детектор открытых глаз

Корицкий Никита

January 16, 2021

1 Введение

Детектирование открытых глаз является актуальной подзадачей в области компьютерного зрения. Для ее решения использовалась неглубокая сверточная сеть, написанная на python с помощью tensorflow. Итоговым решением было использование следующего подхода:

- Частичная(10%) разметка датасета
- Предобучение модели на похожем, но полностью размеченном датасете
- Использование pseudo-labeling из semi-supervised learning

Общая аргументация в защиту данного подхода описаны в главе 2. Итоговый алгоритм и результаты обучения описаны в главах 3 и 4 соответственно.

2 Творческий поиск

Процесс гугления начал выдавать подходы, использующие OpenCV с использованием каскада Хаара. Однако в статье на хабре [1] предупреждали, что это устаревший подох и современные сверточные сети работают гораздо лучше. После этого я наткнулся на статью [2], где для детектора глаз использовалась неглубокая CNN с полносвязным слоем и SVM на конце. Я решил поиграться, чтобы получить первые результаты, но нужен был размеченный датасет. Я быстро нашел датасет [3] и с помощью готового ноутбука из библиотеки tensorflow [4] быстро получил точность в 95-97%. Это был хороший знак, потому что примерно про такую точность на собеседовании говорил Александр.

Дальше нужно было решать вопрос с отсутствием разметки. Я начал гуглить semi-supervised learning. Простым и действенным способом оказалася pseudo-labeling, описанный в статье Lee [5], а в упрощенном варианте в статье на medium [6]. Модифицированная версия подхода [7], которая позволила добиться точности в 99.42% на MNIST с 6% разметкой, легла в основу моего алгоритма.

Суть алгоритма в обучении алгоритма на размеченной выборке, а затем проставление меток на неразмеченном алгоритме из предположения, что алгоритм уже довольно неплохо справляется с задачей. При этом, в начале learning rate домножается на малый коэффициент (потому что мы еще не уверены в результате предсказаний), но с каждой эпохой уверенность растет и коэффициент увеличивается. Раз в несколько батчей

происходит корректировка – обучение на размеченной выборке. В моей задаче всего 2 класса, поэтому данный подход должен был сработать.

Также, я обратил внимание, что изображения в скачанном промеченном датасете CEW были по структуре своей похожи на предоставленные VisionLabs. Поэтому я решил, что можно обучить модель сначала на полностью размеченном датасете, а потом переходить к частично размеченному с меньшим learning rate, обучая только последние слои.

Так или иначе, обучаемый датасет небольшой, поэтому для борьбы с переобучением активно использовались аугментация, батч нормализация и дропауты. Также оказалась важна регуляризация – без нее веса взрывались.

- Горизонтальное отражение
- Поворот на [-0.2π, 0.2π]
- Приближение на [0, 0.1]
- Сдвиг на [0,0.1]

Figure 1: Точность на датасетах CEW и VL

Dataset	Accuracy
CEW train	0.97
CEW val	0.961
VL train	0.997
VL val	0.972

3 Алгоритм и архитектура

За основу архитектуры я взял модель [8], используемую для решения Fasion-MNIST. Но она имела 2 млн. весов и долго обучалась, что не позволяло мне играть с параметрами. Поэтому я сократил все ядра и нейроны в полносвязной части вдвое, что сократило количество весов до 350 тыс., а точность не уменьшилась. В итоге получилась сверточная нейронная сеть, состоящая из 3 слоев сверток с ядрами 32, 64 и 128 и 2 полносвязными слоями на 256 и 128 нейронов. Каждый сверточный слой это:

- Свертка 3x3 с паддингом 1
- Батч нормализация
- Свертка 3x3 без паддинга
- Батч нормализация
- Max Pooling 2x2

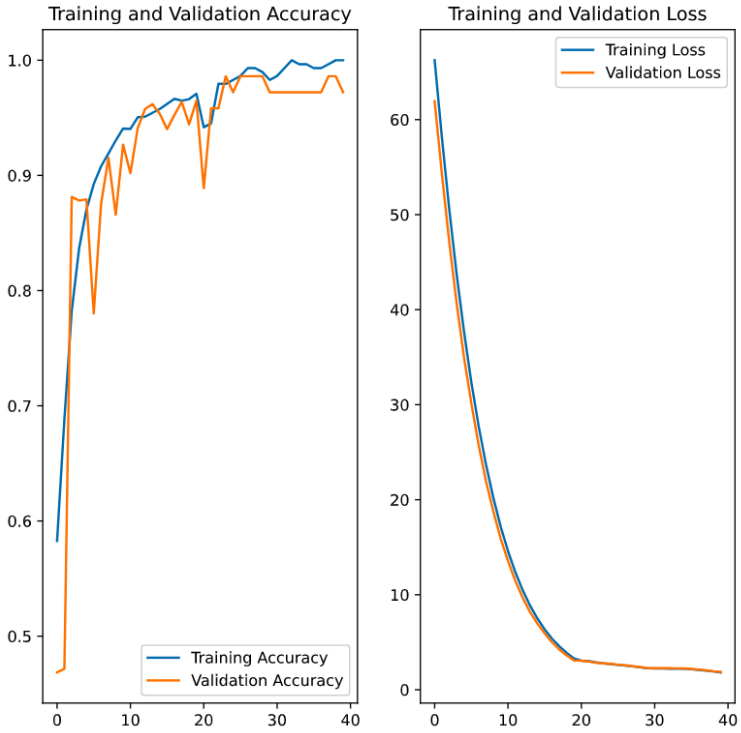
Каждый слой имел L2 регуляризацию с $\lambda = 0.1$, а после каждого слоя использовался дропаут с вероятностями от 0.2 до 0.4. На выходе сети 1 нейрон, знак значения кототорого определяет класс. Оптимизатор – Adam, функция активации – relu (кроме последнего нейрона), лосс функция – бинарная кросс энтропия работающая с логитами напрямую для численной стабильности.

Figure 2: Простые и сложные изображения для алгоритма. Над фото значения выходного нейрона. Если оно отрицательное, то алгоритм детектирует глаза как закрытые, если положительное, то открытые. Зеленый цвет означает, что алгоритм верно угадал, красный – ошибся.

Сеть предобучается на CEW датасете (3877 обучающих и 969 тестовых изображений (24,24,1)) в течение 20 эпох. Затем первые 2 слоя (с ядрами 32 и 64) замораживаются. Модель обучается на неразмеченном датасете с псевдо разметкой с learning rate в 10 раз меньшим, чем для размеченного датасета. На этом этапе несколько раз в эпоху происходит корректировка модели на размеченном датасете (из вручную размеченного датасета от VL) с исходным learning rate. С каждой новой эпохой learning rate для обучения с псевдо разметкой линейно растет.

Входные изображения подвергались обработке. Они нормировались на интервал [-1,1],

Figure 3: Графики обучения



а также случайной аугментации вида:

4 Результаты

Training и validation точность для обоих датасетов предоставлена на Рис 1. Сложные и простые для алгоритма картинки предоставлены на Рис 2. Видно, что алгоритм плохо спраляется примерно там же, где может ошибиться и человек. Отсюда явные ограничения подхода с частичной разметкой. В начале я позволял себе размечать фото, в которых не уверен, но такой подход давал низкий (88%) результат на VL датасете. Затем я оставлял только изображения, в которых уверен, а значит алгоритм не мог обучаться на трудных примерах.

Графики обучения изображены на Рис 3. Видно, что происходит изменение характера обучения после 20 эпохи, когда происходит переход на semi-supervised learning.

EER = 0.0196

5 Выводы

Предложенный метод позволяет быстро обучить несложную модель, которая добьется хорошей точности на предоставленном датасете. Мне повезло, что найденный мною размеченный датасет оказался похожим на предоставленный VL. Тем не менее, используемый метод работал с точностью около 90% на частично (и плохо) размеченных данных без предобучения.

References

[1] Обучение opencv каскада Хаара / Хабр. https://habr.com/ru/post/208092/. (Accessed on 01/13/2021).

[2] Mingxin Yu, Xiaoying Tang, Yingzi Lin, David Schmidt, Xiangzhou Wang, Yikang Guo, and Bo Liang. An eye detection method based on convolutional neural networks and support vector machines. *Intelligent Data Analysis*, 22(2):345–362, March 2018.

[3] The closed eyes in the wild (cew) dataset. http://parnec.nuaa.edu.cn/_upload/tpl/02/db/731/template731/pages/xtan/ClosedEyeDatabases.html. (Accessed on 01/13/2021).

[4] Image classification | tensorflow core. https://www.tensorflow.org/tutorials/images/classification. (Accessed on 01/13/2021).

[5] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[6] Pseudo-labeling to deal with small datasets — what, why & how? | by anirudh shenoy | towards data science. https://towardsdatascience.com/pseudo-labeling-to-deal-with-small-datasets-what-why-how-fd6f903213af. (Accessed on 01/14/2021).

[7] Github - peimengsui/semi_supervised_mnist. https://github.com/peimengsui/semi_supervised_mnist. (Accessed on 01/14/2021).

[8] Classifying fashion with a keras cnn (achieving 94% accuracy) — part 1 | by manish bhobé | medium. https://medium.com/@mjbhobe/classifying-fashion-with-a-keras-cnn-achieving-94-accuracy-part-1-1ffcb7e5f61a. (Accessed on 01/15/2021).