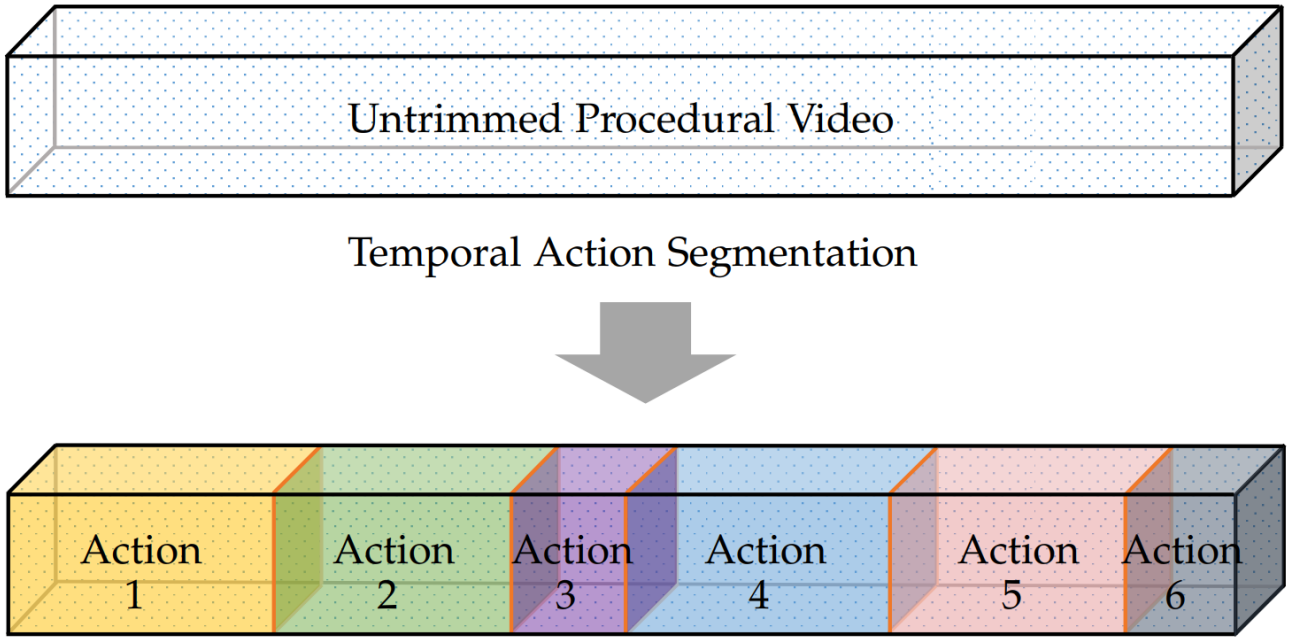
Мультимодальное предобучение трансформеров для распознавания действий в видео-инструкциях



Научный руководитель: Борис Зимка
Студент: Чувиляева Ирина
Кафедра Интеллектуальной Обработки Документов

Постановка задачи: исследовать влияние мультимодального предобучения при распознавании действий в видео-инструкциях с помощью трансформеров

- Разобраться с литературой, выбрать подходящий baseline
- Обучить одномодальную трансформер-подобную модель на задаче Action Segmentation без pretrain
- Выбрать мультимодальную трансформер-подобную модель
- Попытаться самостоятельно предобучить мультимодальную модель или взять уже готовый чекпоинт
- Выделить часть мультимодальной модели, которая обрабатывает видео
- Провести fine-tune видео-части мультимодальной модели
- Сравнить результаты на целевом датасете

План работы

- ноябрь – декабрь 2022: чтение литературы, постановка целей
- январь – март 2023: обучение одной из известных моделей без pretrin, с подсчётом собственных векторов признаков для видео
- апрель – июнь 2023: фиксация одномодальной модели, замеры качества.
- сентябрь – октябрь 2023: выбор модели с несколькими модальностями
- ноябрь 2023 – февраль 2024: предобучение мультимодальной модели
- март – апрель 2024: выделение видео-части мультимодальной модели, адаптация под целевой датасет
- май 2024: фиксация и тюнинг мультимодальной модели, замеры качества

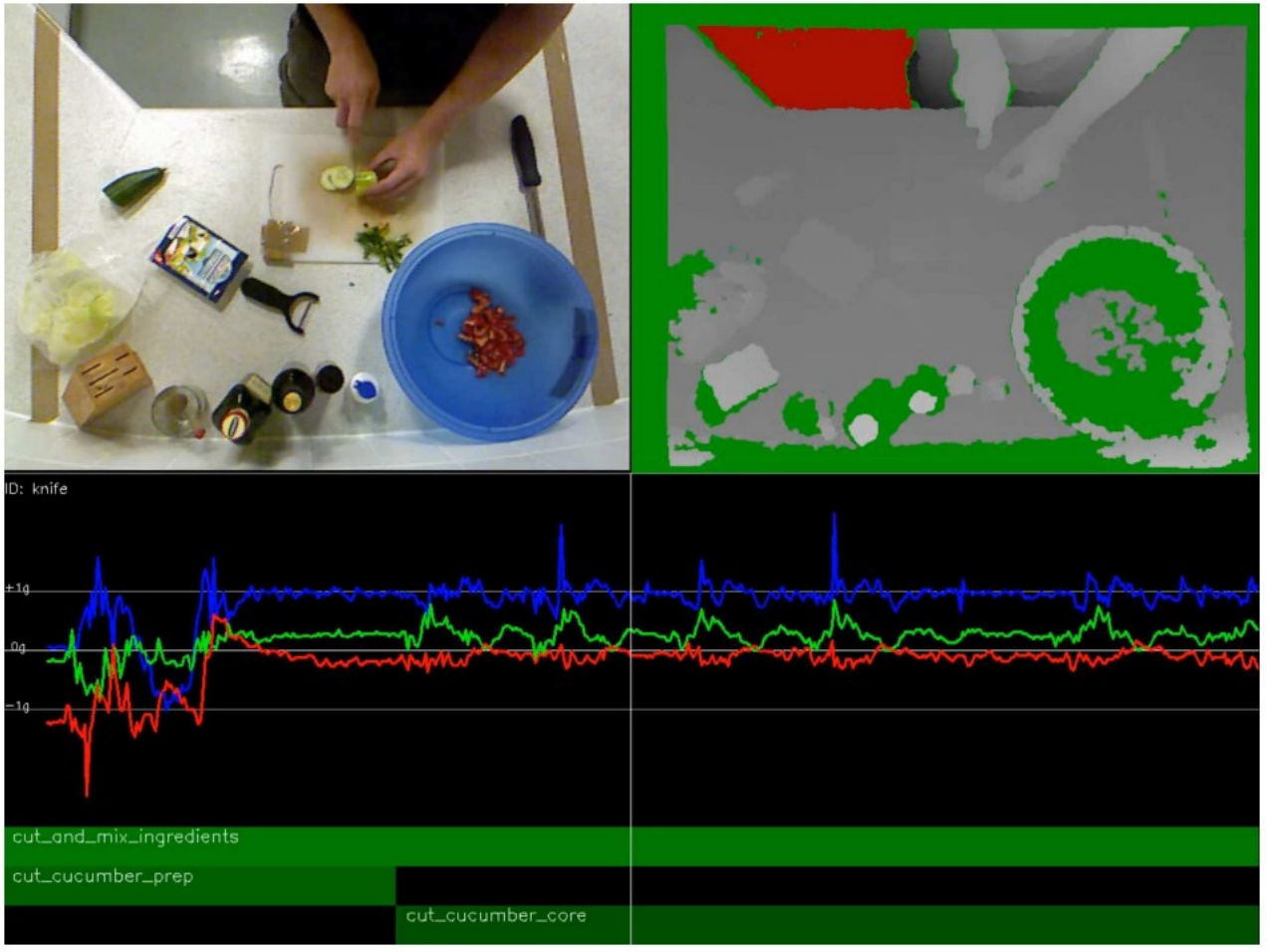
  IV контроль, предзащита
- июнь 2024: подготовка итогового текста работы

  защита

Датасеты

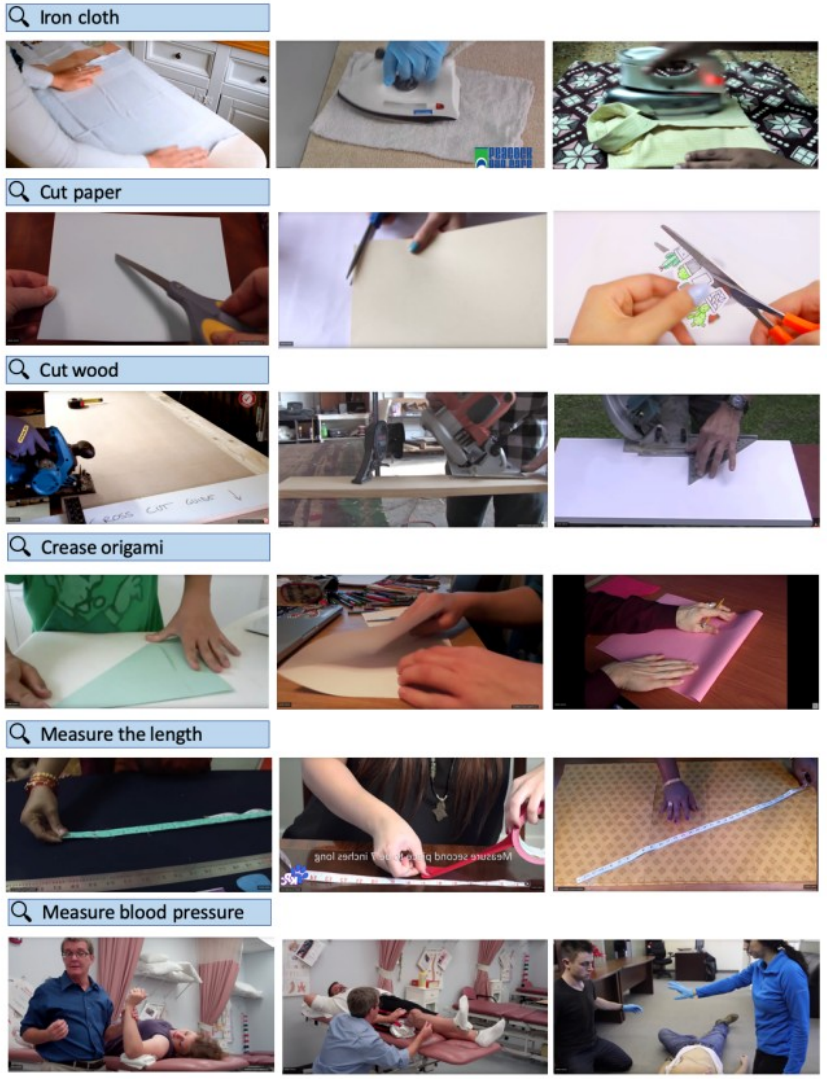
- 50Salads: целевой датасет. На нём проверяем работоспособность моделей и делаем все итоговые замеры.
- HowTo100M: датасет для мультимодального pretrain. В обучении используются векторы признаков, полученные из S3D. Для обучения S3D применялся этот датасет.

50 Salads



- 50 видео, на которых 25 человек готовят 2 различных салата в случайной последовательности
- 4.5 часов видео
- 30 fps
- 17 различных действий, а так же классы action_start и action_end
- от первого лица
- 5 сплитов для кросс-валидации

Датасет для мультимодального pretrain: HowTo100M

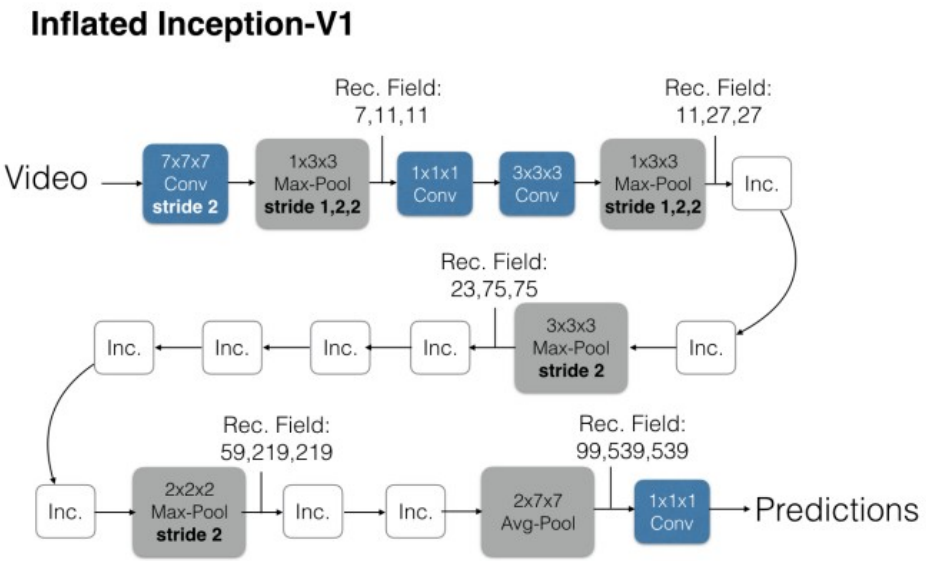


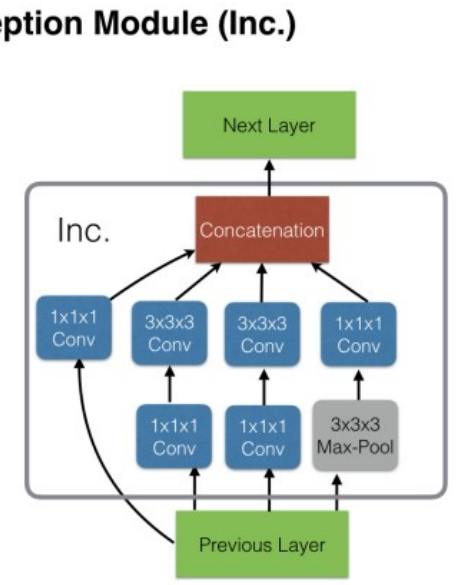
- 1.2M видео, в которых пытаются научить выполнять некоторое сложное действие, собраны с Youtube
- 15 лет видео
- 23.6K действий
- Положение наблюдателя может варироваться
- Шумные субтитры в силу характера сбора данных
- В силу колоссального объёма использованы уже извлечённые признаки. Нейросеть для извлечения признаков: S3D.
- S3D использует 10 fps во время обучения и 30 fps во время извлечения векторов признаков, так получается 1 вектор в секуду

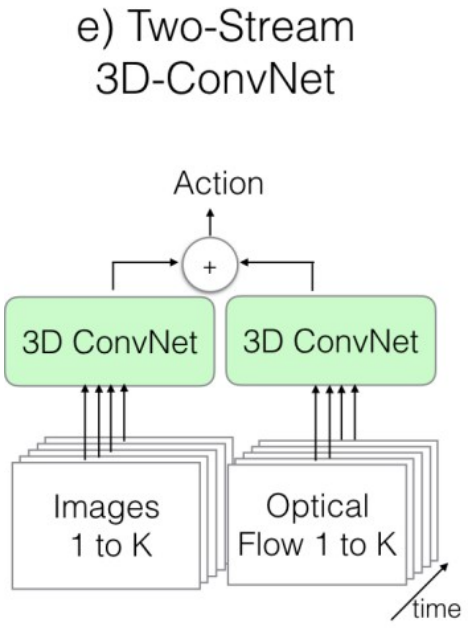
Описание одномодального baseline

- **backbone: I3D**, предобученная на датасете Kinetics (64 fps)
  - Не учим сами, а используем готовые веса из pytorchvideo
  - Проходимся по кадрам видео окном размера 21, как авторы ASFormer, и подаём кадры в сеть. На выходе получаем векторы признаков размерности 2048 для каждого кадра из видео
- основная модель: **ASFormer**
  - Принимает на вход векторы признаков, использует вплоть до 512 соседних векторов признаков
  - На выходе получаем класс действия

Backbone: I3D





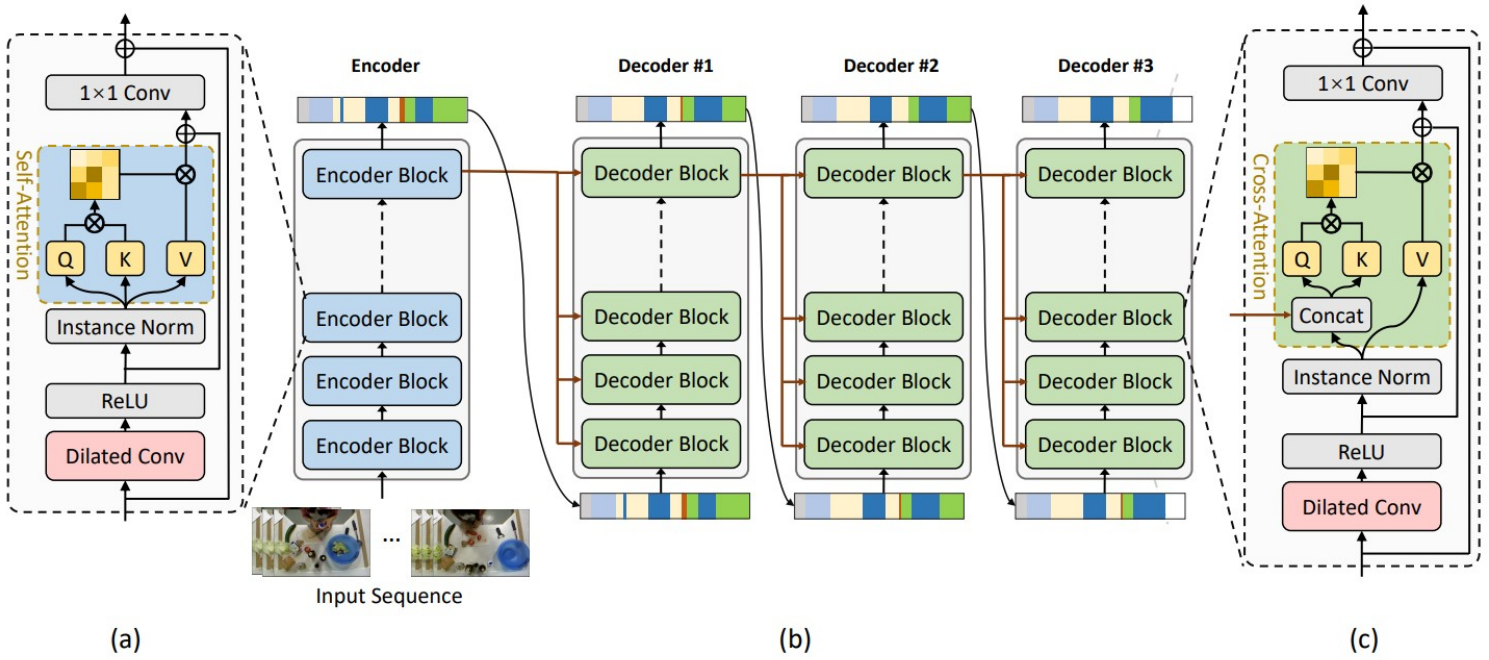


- “Раздуваем” веса из Inception-V1, предобученной на ImageNet во временном пространстве.
- Сетей на самом деле 2: для кадров и для оптического потока. Итоговое предсказание получается усреднением
- 25M параметров

ASFormer: Transformer for Action Segmentation

Особенности:

- dilated temporal convolution вместо линейного слоя в feed forward блоках
- hierarchical pattern в attention учитываются только кадры на расстоянии до 2, 4, 8 … 512 – размер окна и dilation свёртки растут с номером блока.
- По 9 блоков в энкодере и декодерах

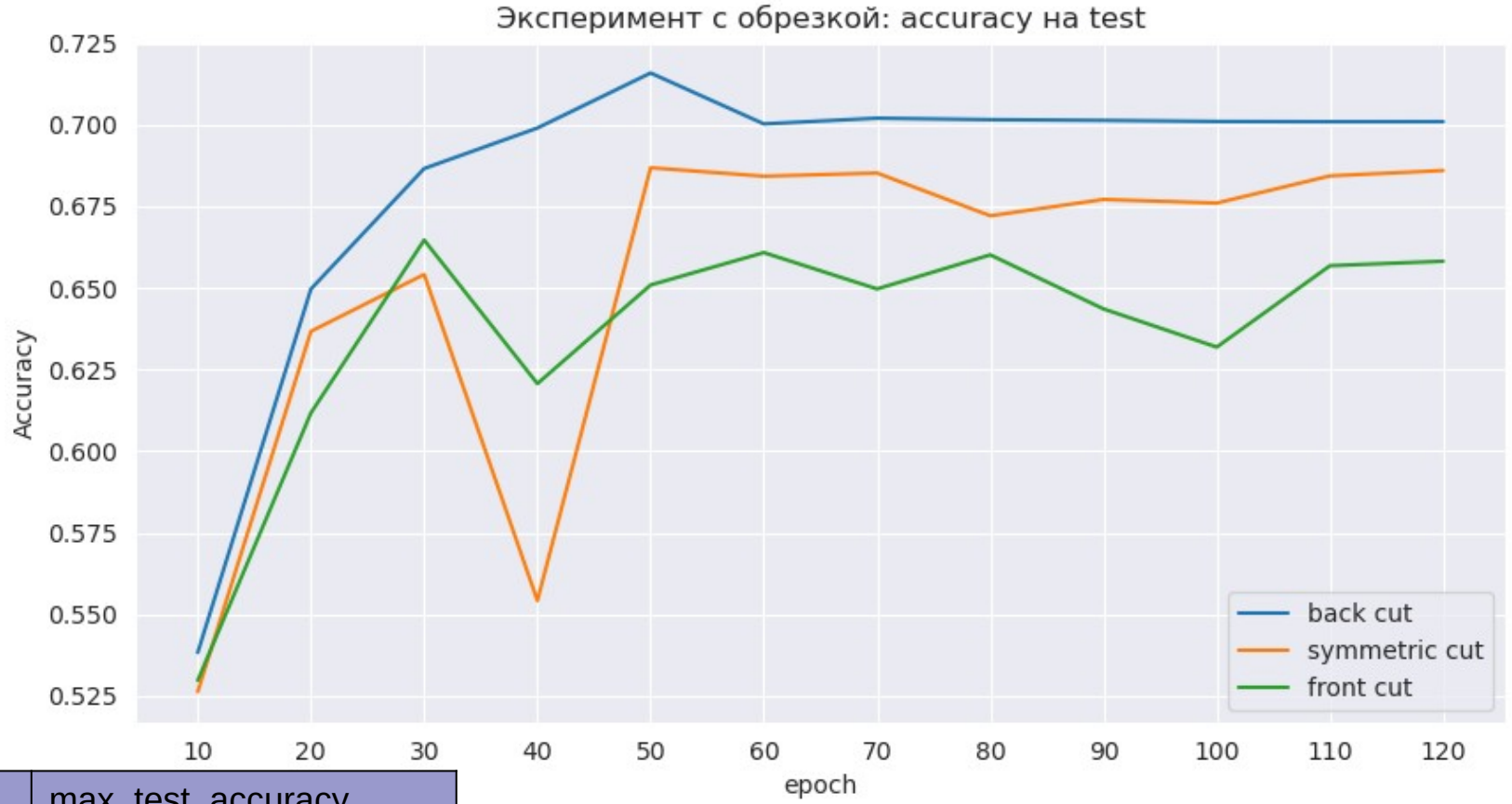


- 1 энкодер и 3 декодера. Последовательно улучшаем предсказания
- В декодерах при подсчёте key и query конкатенируем результаты предыдущего слоя с предсказаниями предыдущего энкодера или декодера. Value берётся просто из предыдущего слоя

Прогон I3D на 50 Salads: результаты

- При работе с датасетом было найдено несколько повреждений, поэтому
  - Открываем только через VideoCapture из opencv
  - Пропускаем повреждённые кадры.
- После прогона через модель получаются признаки размером на 8 больше, чем у авторов ASFormer. Эта цифра является постоянной для всех видео в датасете.
- Для выбора правильной стратегии обрезки потребовался дополнительный эксперимент

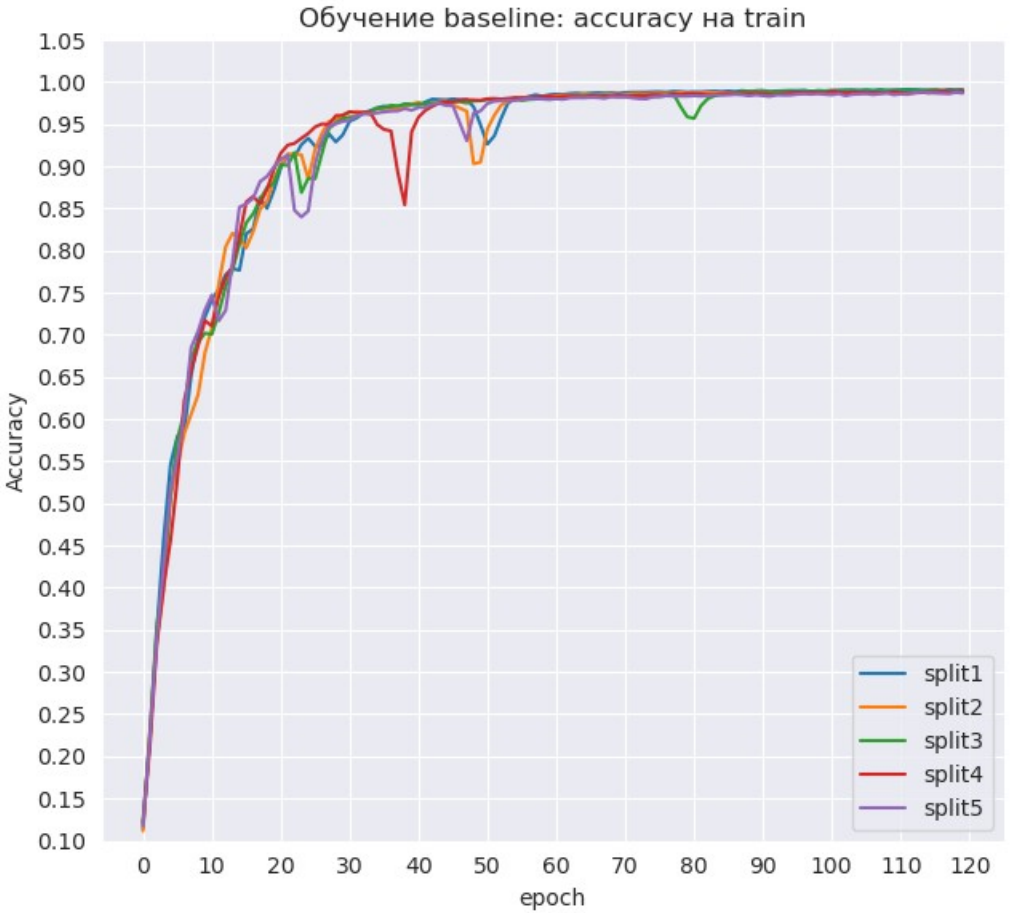
Эксперимент с обрезкой видео

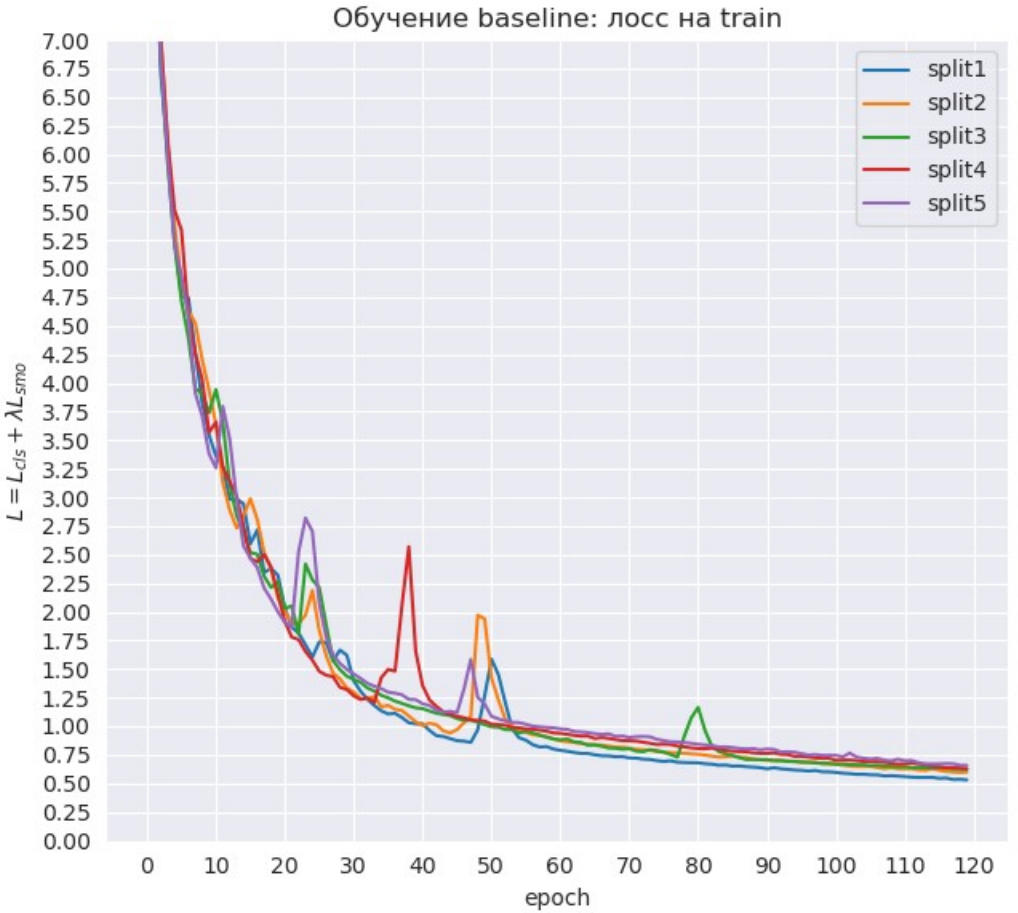


cut	max_test_accuracy
front	0.665
symmetric	0.687
back	0.716

Вывод: берём обрезку сзади

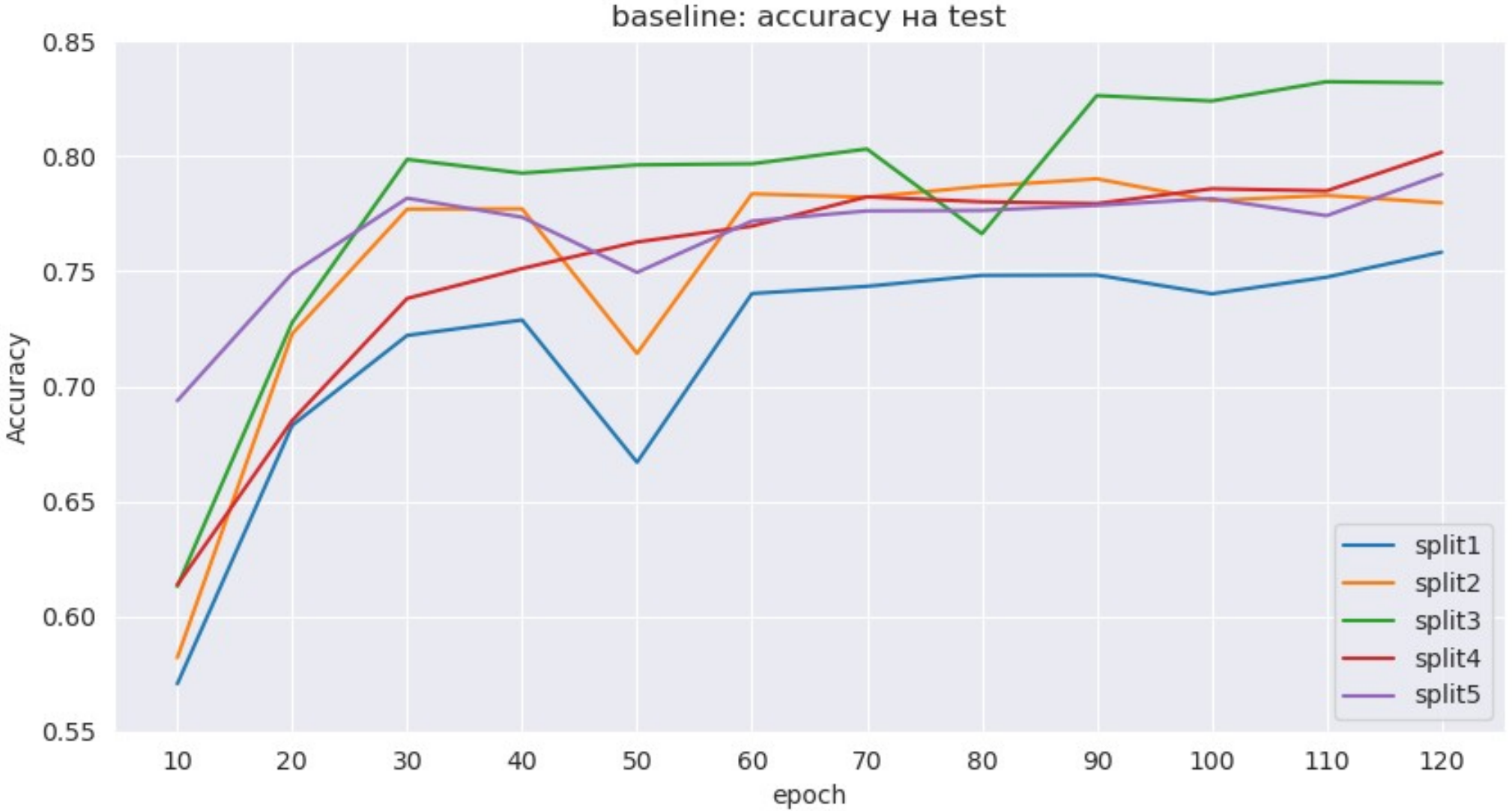
ASFormer: обучение с кросс-валидацией





Функция потерь – взвешенная сумма кросс-энтропии и сглаживающего по времени квадратичного лосса с коэффициентом $\lambda = 0.25$

Baseline: результаты



Baseline: сравнение результатов с авторскими

При создании собственных векторов признаков с помощью I3D из pytorchvideo

split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_accuracy	got_on_epoch
1	0.758	66.2	0.751	0.728	0.623	0.758	120
2	0.780	75.6	0.808	0.760	0.708	0.790	90
3	0.832	77.3	0.841	0.831	0.784	0.832	110
4	0.802	72.5	0.793	0.784	0.700	0.802	120
5	0.792	70.5	0.788	0.749	0.680	0.792	120
среднее	0.793	72.4	0.796	0.770	0.699	0.795	-

Результат из статьи:

0.856	79.6	0.851	0.834	0.760	

При использовании авторских векторов признаков

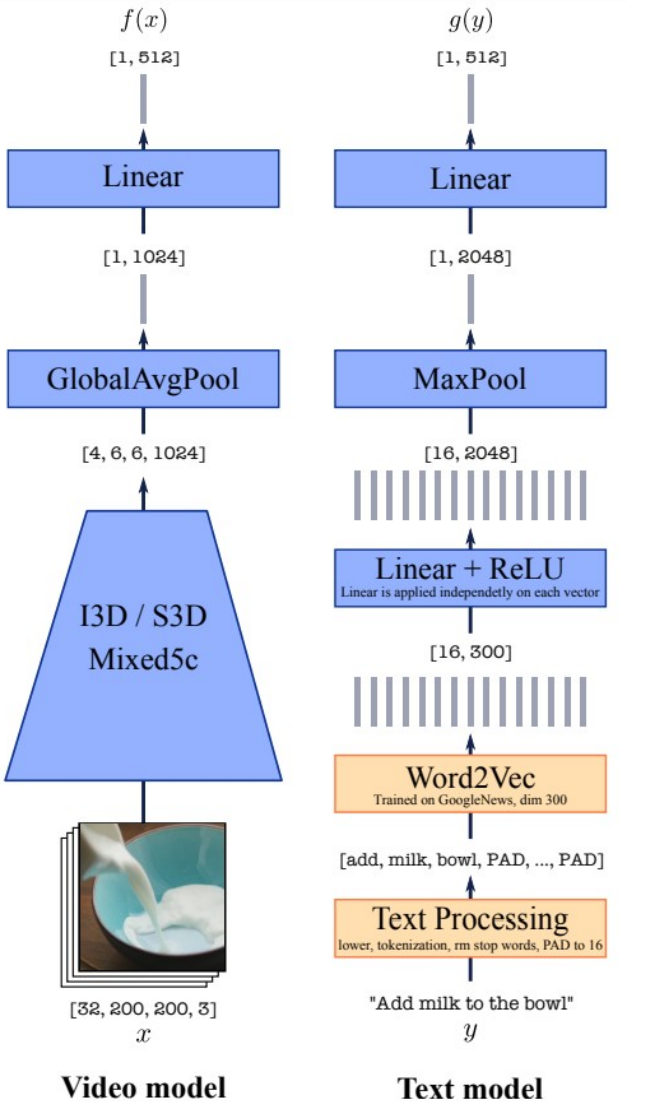
split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_accuracy	got_on_epoch
1	0.809	75.4	0.807	0.794	0.700	0.809	120
2	0.885	74.6	0.816	0.794	0.739	0.893	100
3	0.849	78.9	0.840	0.835	0.769	0.858	110
4	0.862	80.1	0.859	0.835	0.772	0.866	100
5	0.865	80.0	0.866	0.856	0.761	0.865	110
среднее	0.854	77.8	0.838	0.823	0.748	0.858	-

Описание мультимодальной модели

После рассмотрения нескольких вариантов, выбрана мультимодальная модель **VideoCLIP**

- Благодаря contrastive learning, из модели проще выделить видео-часть
- Авторы смогли адаптировать pretrain на датасете HowTo100M для задач
  - Text Video Retrieval (YouCook2, MSR-VTT, DiDeMo)
  - VideoQA (MSR-VTT, 5 кандидатов на запрос)
  - Action Segmentation (COIN)
  - Action Step Localization (CrossTask)
- Возможны и zero-shot transfer learning и fine-tuning, что позволяет провести больше экспериментов
- В качестве backbone используется нейросеть S3D
  - Она схожа с backbone одномодальной модели (I3D)
  - Имеются предпосчитанные на датасете HowTo100M векторы признаков

Backbone мультимодальной модели: S3D



S3D – это сепарабельная по времени I3D, обученная на HowTo100M с некоторыми отличиями:

- Новый лосс: MIL-NCE

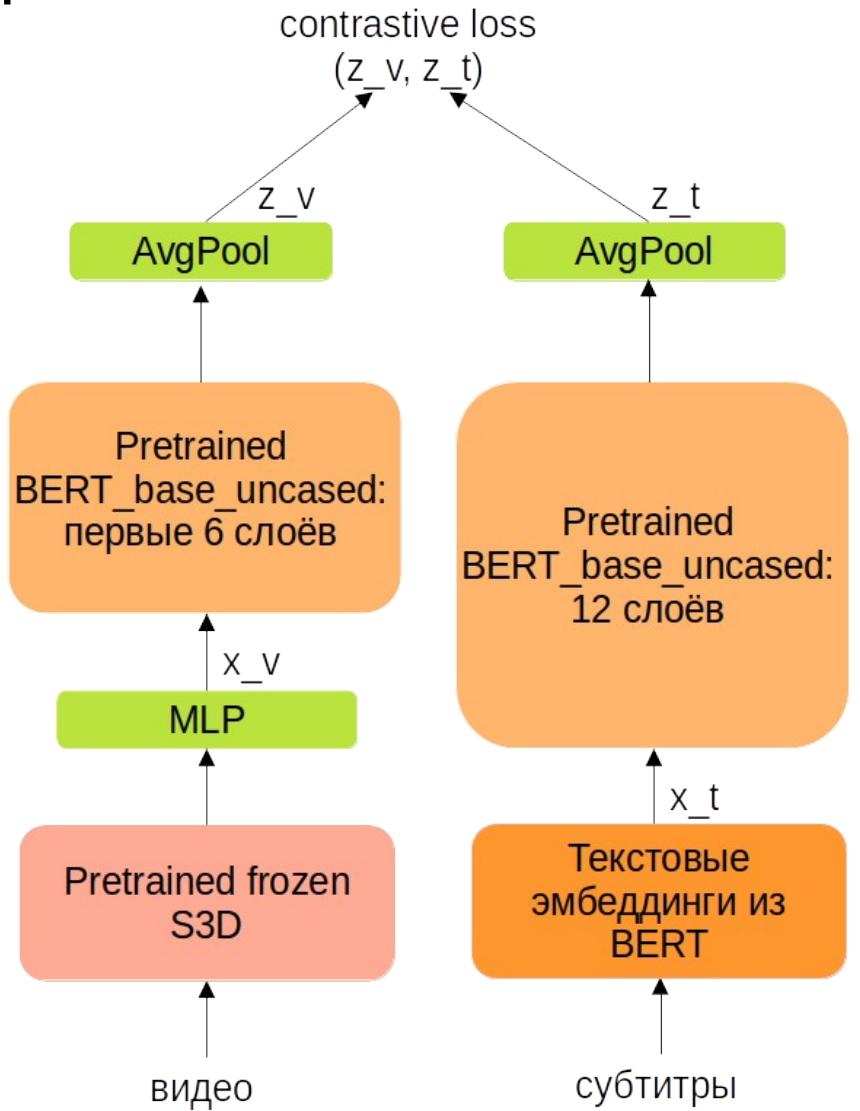
$$\max_{f,g} \sum_{i=1}^{n} \log \left(\frac{\sum_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)}}{\sum_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)} + \sum_{(x',y')\sim\mathcal{N}_i} e^{f(x')^\top g(y')}} \right)$$

Позитивные примеры – это K = 5 ближайших по времени, а негативные – это 512 пар сэмплированных из всего батча.

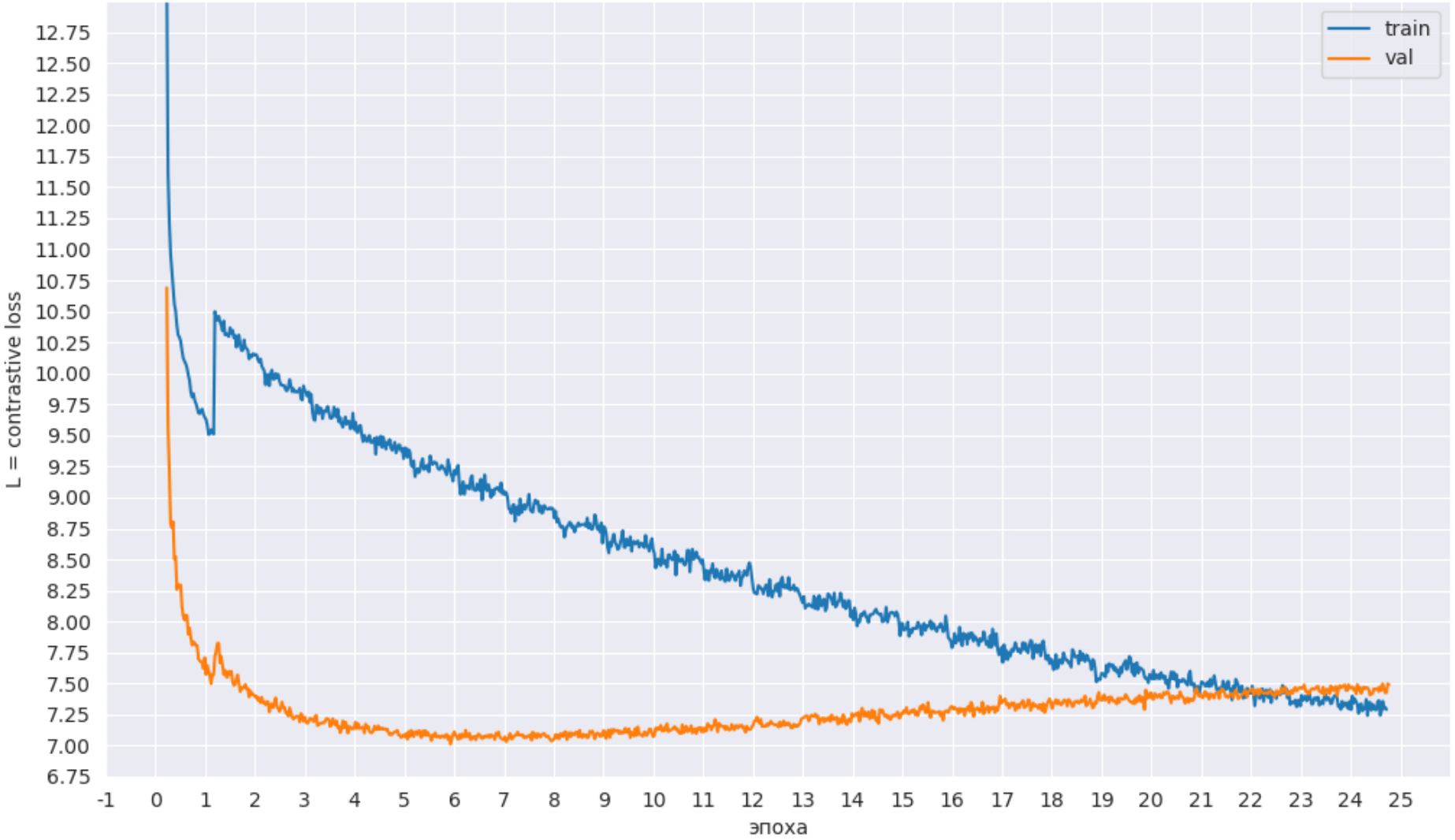
- Модель обучается без разметки.
- Дополнительно используется простой word2vec для текстовой модальности.
- Другой оптимизатор (Adam).

Мультимодальная модель: VideoCLIP

- Модель состоит из двух отдельных BERT для видео и для текста.
- Contrastive loss – это сумма NCE-лоссов для пар (z_v, z_t) и (z_t, z_v)
- Алгоритм для поиска сильных негативных примеров:
  - Выбирает C = 8 случайных видео во всём пространстве
  - Выбирает k = 32 случайныхидео равноудалённых в вокруг каждого из C в общий кластер (батч). Далее обучение проходит внутри батча.
  - Одно видео даёт 16 пар сегментов.
- Разрешено пересечение сегментов во времени. Сегменты удлинены



Pretrain VideoCLIP на одном GPU

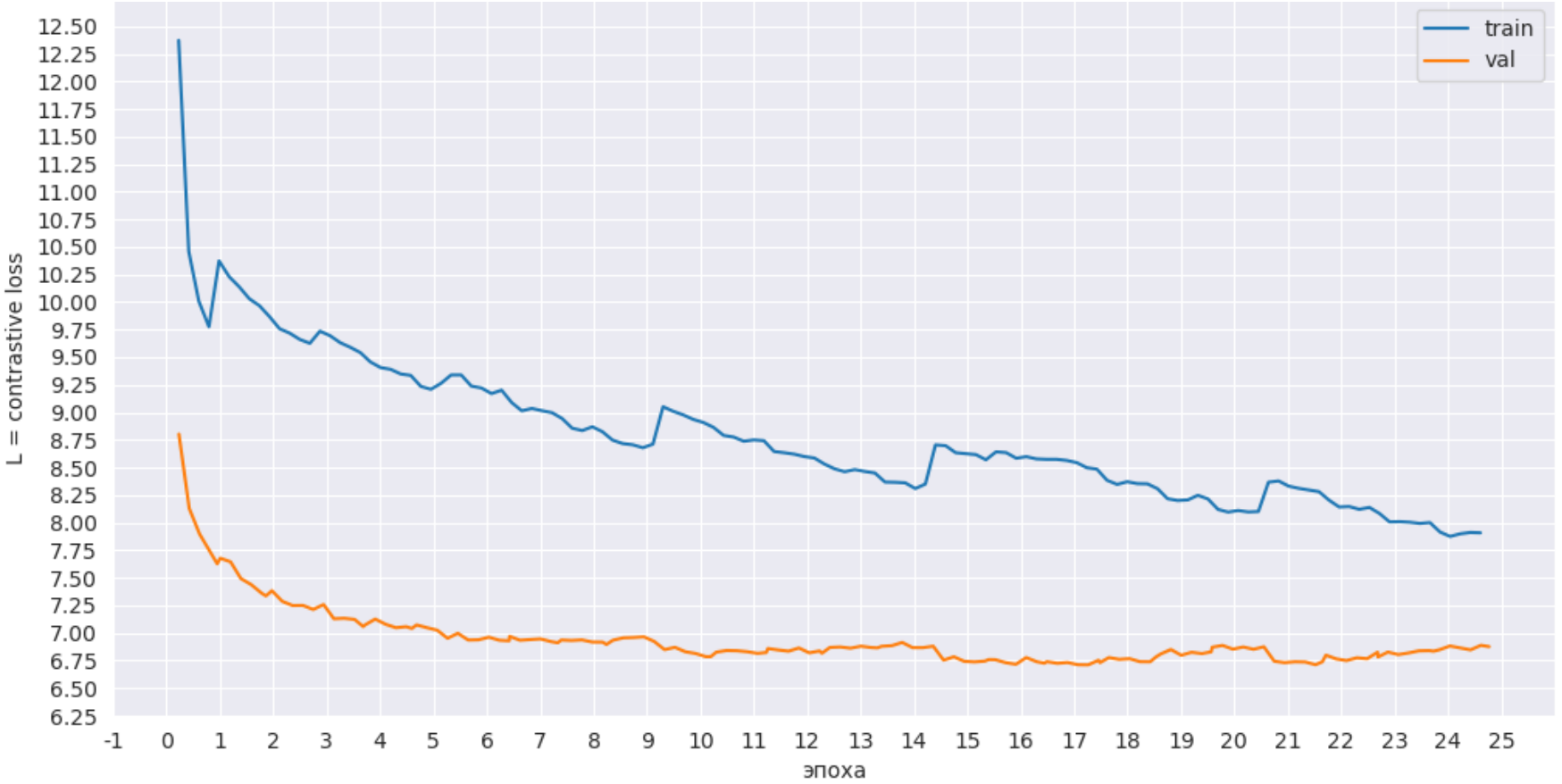


Почему эксперимент не удался?

- Основная причина: **использование C = 1 кластеров (батчей)**
- При обучении на 1 GPU алгоритм сэмплирования негативных примеров выбирает C = 1 случайное видео, строит 1 кластер и проводит обучение внутри кластера
    - Используем меньшую обучающую выборку внутри эпохи
    - Случайный кластер может быть сильно смещён, шаги модели становятся большими и хаотичными
- Авторы VideoCLIP писали код опираясь на библиотеку fairseq и предполагая наличие кластера с ОС linux из 8 GPU типа A100. Считали, что на 1 GPU будет обрабатываться 1 батч (кластер) из 32 видео по 16 сегментов.
    - С текущими возможностями видео-карт на 1 GPU можно поместить не более 2 батчей. Уменьшение размера батча ведёт к меньшему числу негативных примеров и к деградации модели
    - Чтобы распараллелить на несколько GPU требуется “перевести” fairseq на ОС windows

Вывод: для предобучения модели нужны специфичные и дорогостоящие ресурсы.

Pretrain VideoCLIP на семи GPU



* К несчастью, идеальный сервер найти не удалось. На найденной машине пришлось пожертвовать 1 GPU и несколько раз прервать обучение в силу совместного доступа. Требования модели были очень серьёзными и стоит радоваться, что удалось получить некоторый результат

Выделение видео-части VideoCLIP для задачи Action Segmentation

- Пропускаем названия классов действий через текстовую часть модели и получаем их представления в латентном пространстве h_t
- Убираем AveragePooling в конце обработки видео. Теперь модель покадровые векторы признаков h_v
- Используем скользящее окно размером 32 и шагом 16.
- Пересчитываем скалярные произведения h_v со всеми векторами h_t и выбираем максимум как предсказание
- НЕ предсказываем фоновый класс

Zero-shot перенос: запускаем описанный алгоритм на валидации

Fine-tune: учим модель с contrastive loss для пар (h_v, h_t) 8 эпох.

Результаты zero-shot переноса

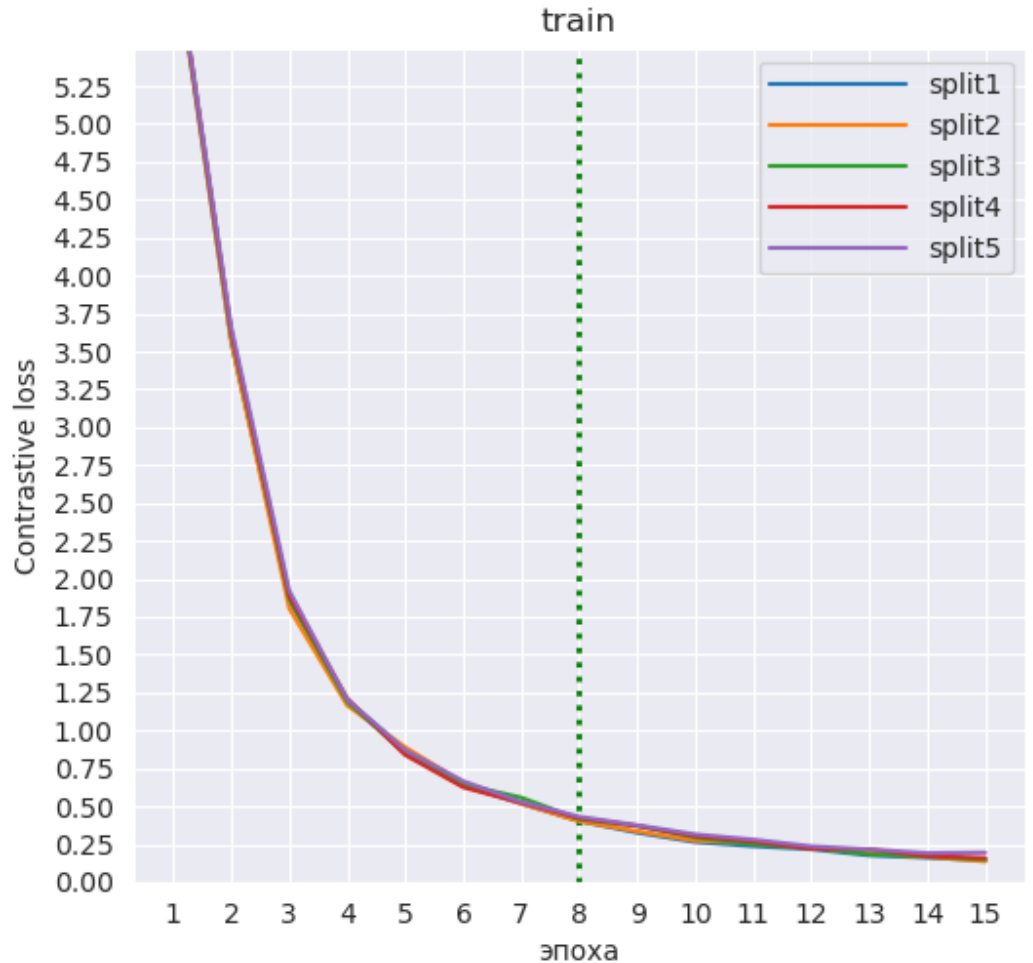
Для самостоятельно обученной модели

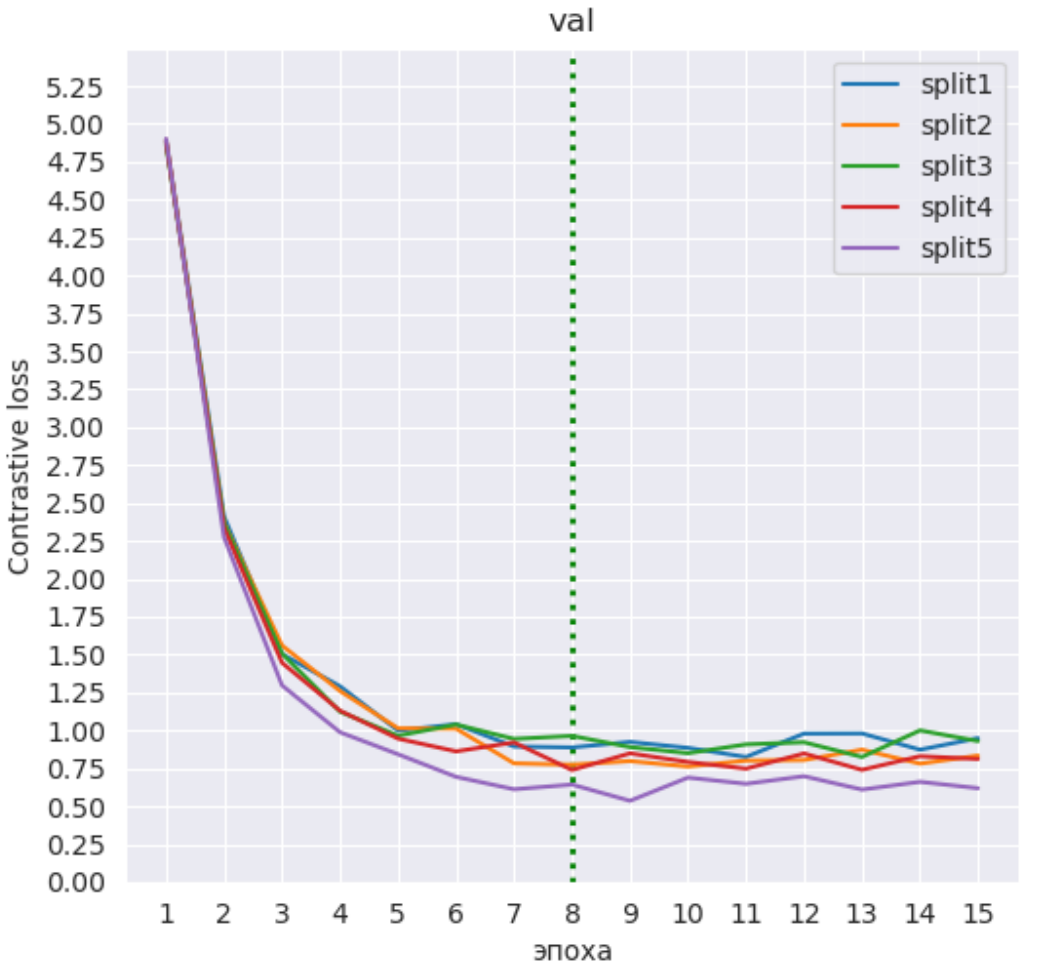
split	end_accuracy	edit_dist	f1@10	f1@25	f1@50
1	0.296	278.1	0.127	0.075	0.042
2	0.287	273.9	0.125	0.079	0.029
3	0.323	261.3	0.142	0.093	0.049
4	0.300	269.6	0.151	0.084	0.041
5	0.264	267.9	0.138	0.091	0.035
среднее	0.294	270.2	0.137	0.084	0.039

Для модели с общедоступными весами

split	end_accuracy	edit_dist	f1@10	f1@25	f1@50
1	0.413	230.1	0.150	0.094	0.044
2	0.469	201.4	0.159	0.104	0.053
3	0.483	197.7	0.184	0.127	0.071
4	0.483	196.9	0.184	0.131	0.072
5	0.437	203.2	0.182	0.123	0.081
среднее	0.457	205.9	0.172	0.116	0.064

Дообучение собственной модели на целевом датасете





8 эпох – длительность дообучения, предложенная авторами для датасета COIN

Результаты самостоятельно обученной модели

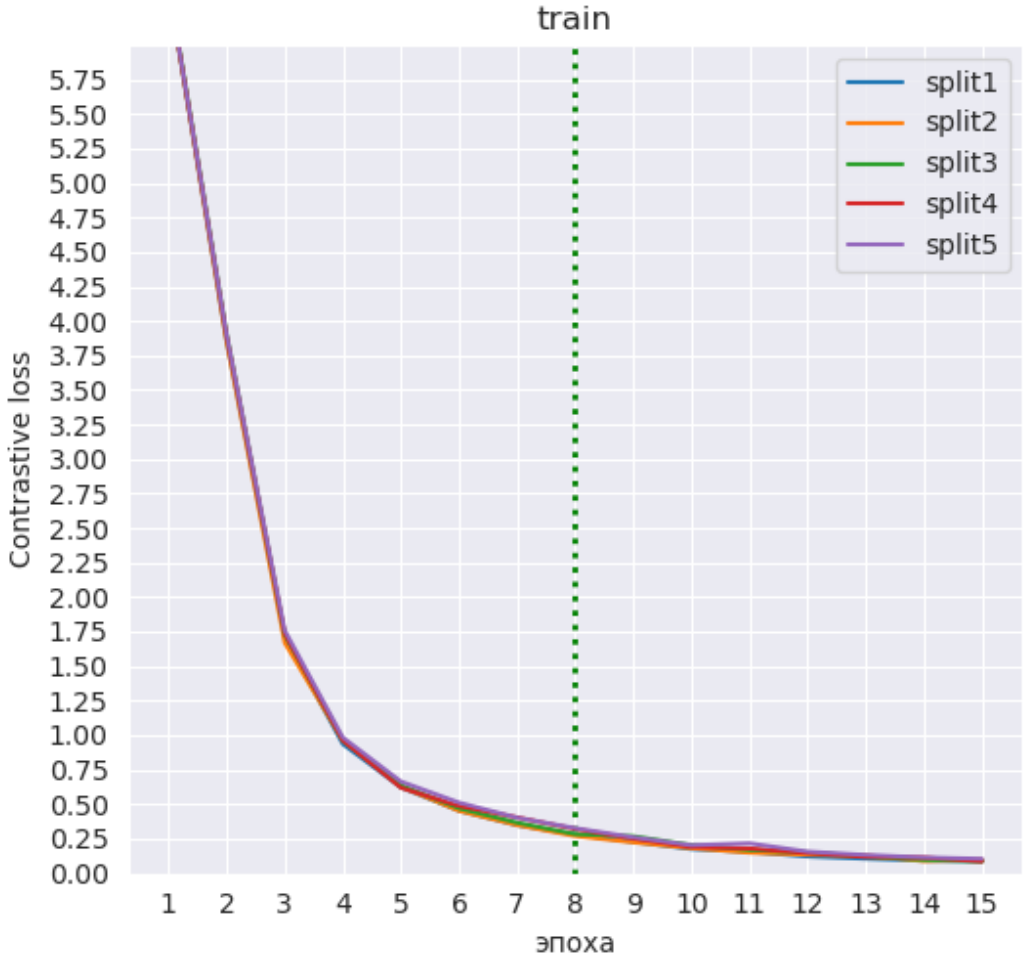
После 8 эпох дообучения

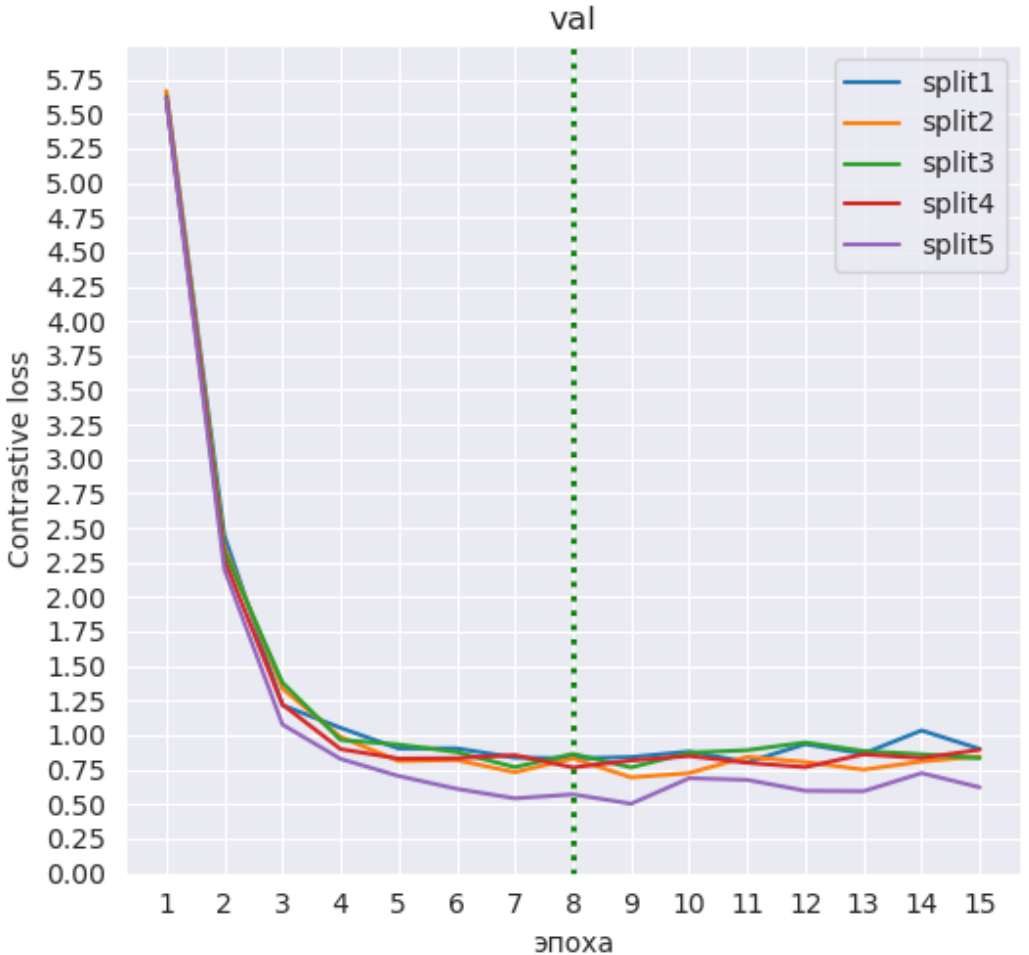
split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_loss	epoch
1	0.837	58.6	0.486	0.455	0.381	0.888	8
2	0.791	79.2	0.465	0.429	0.346	0.774	8
3	0.753	94.2	0.496	0.446	0.347	0.944	7
4	0.799	77.1	0.503	0.463	0.384	0.739	8
5	0.816	65.7	0.498	0.443	0.388	0.612	7
среднее	0.799	75.0	0.490	0.447	0.369	0.791	-

После 15 эпох дообучения

split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_loss	epoch
1	0.802	76.2	0.485	0.453	0.366	0.825	11
2	0.807	72.9	0.498	0.458	0.360	0.762	10
3	0.812	71.4	0.515	0.467	0.392	0.825	13
4	0.799	77.1	0.503	0.463	0.384	0.739	8
5	0.837	58.6	0.486	0.455	0.381	0.536	9
среднее	0.811	71.2	0.497	0.459	0.377	0.737	-

Дообучение общедоступной модели





8 эпох – длительность дообучения, предложенная авторами для датасета COIN

Результаты общедоступной модели

После 8 эпох дообучения

split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_loss	epoch
1	0.788	82.4	0.549	0.503	0.405	0.830	8
2	0.806	73.7	0.500	0.462	0.401	0.730	7
3	0.801	76.6	0.602	0.543	0.480	0.767	7
4	0.802	75.2	0.562	0.532	0.476	0.765	8
5	0.851	53.2	0.541	0.525	0.449	0.541	7
среднее	0.810	72.2	0.551	0.513	0.442	0.727	-

После 15 эпох дообучения

split	end_accuracy	edit_dist	f1@10	f1@25	f1@50	best_loss	epoch
1	0.813	72.3	0.593	0.552	0.463	0.803	11
2	0.810	72.2	0.540	0.540	0.430	0.693	9
3	0.814	71.1	0.565	0.531	0.463	0.765	9
4	0.802	75.2	0.562	0.532	0.476	0.765	8
5	0.842	57.3	0.579	0.559	0.486	0.502	9
среднее	0.816	69.6	0.568	0.543	0.464	0.706	-

Сводная таблица результатов

метод	среднее accuracy
ASFormer по версии официальной статьи	0.856
ASFormer на авторских векторах признаков	0.854
baseline: ASFormer на векторах признаков из I3D	0.793
VideoCLIP: мультимодальный pretrain, затем zero-shot перенос	0.294
VideoCLIP: общедоступные веса после мультимоального pretrain, затем zero-shot перенос	0.457
VideoCLIP: мультимодальный pretrain, затем дообучение 8 эпох	0.799
VideoCLIP: мультимодальный pretrain, затем дообучение 15 эпох	0.811
VideoCLIP: общедоступные веса после мультимоального pretrain,, затем дообучение 8 эпох	0.810
VideoCLIP: общедоступные веса после мультимоального pretrain,, затем дообучение 15 эпох	0.816

Выводы

- Удалось приблизиться к авторскому качеству с помощью нейросети ASFormer. Это позволило получить сильный baseline для последующей работы
- Воспроизведение полного обучения мультимодальных моделей остаётся сложной ресурсоёмкой задачей. Даже небольшие отклонения от авторской процедуры могут привести к серьёзным последствиям
- При сравнении с мультиомодальным VideoCLIP видно, что zero-shot справляется с задачей ActionSegmentation значительно хуже. Вероятно, это связанно с выбором специфического датасета, в котором есть близкие классы (cut cucumber и peel cucumber) и классы, которые можно условно отнести к background (action start и action end).
- При этом после fine-tune получается модель, способная соперничать с одномодальной специально обученной на целевой датасет архитектурой. Это говорит о способности трансформер-подобных моделей адаптироваться к новым задачам.
- Если по каким-то причинам авторское мультимодальное предобучение не удаётся повторить полностью, может помочь разумная адаптация процедуры дообучения.
- Гипотеза об улучшении качества модели за счёт мультимодального предобучения подтвердилась, но следует уточнить, что улучшения могут быть незначительными по сравнению с затраченными ресурсами, поэтому на практике следует применять данный метод с осторожностью.
- В случае нехватки ресурсов разумной альтернативой остаётся использование общедоступных предобченных весов.

Литература

- “Temporal Action Segmentation: An Analysis of Modern Technique” - Guodong Ding, Fadime Sener, Angela Yao
- “Multimodal Learning with Transformers: A Survey” - Peng Xu, Xiatian Zhu, and David A. Clifton
- “ASFormer: Transformer for Action Segmentation” - Fangqiu Yi, Hongyu Wen, Tingting Jiang
- “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset” - Joao Carreira, Andrew Zisserman
- “VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding” - Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, Christoph Feichtenhofer
- “End-to-End Learning of Visual Representations from Uncurated Instructional Videos” - Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev1, Josef Sivic, Andrew Zisserman
- “Combining embedded accelerometers with computer vision for recognizing food preparation activities” - Sebastian Stein and Stephen J. McKenna
- “HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips” - Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, Josef Sivic

- “PyTorchVideo: A Deep Learning Library for Video Understanding” - Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross B. Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, Christoph Feichtenhofer
- “Fairseq: A Fast, Extensible Toolkit for Sequence Modeling” - Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, Michael Auli
- “Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification” - Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu and Kevin Murphy

Спасибо за внимание!