Отчёт по классификации изображений (MNIST)

Введение

Цель данной работы – продемонстрировать применение классических (SVM) и нейросетевых (MLP) методов машинного обучения для классификации изображений рукописных цифр из датасета MNIST. В процессе экспериментов мы:

1. Выполнили подготовку данных (объединение тренировочной и тестовой выборок, нормализация, разбиение на обучающую и валидационную части).
2. Настроили **SVM**
3. Построили **MLP** (полносвязную нейронную сеть)

1. Подготовка данных

Описание этапа

- **Загрузка датасета MNIST**: данные (x_train, y_train), (x_test, y_test) объединены в общий массив, чтобы затем вручную разделить их в соотношении 80/20.
- **Нормализация**: пиксели, имеющие изначально значения в диапазоне [0..255], переведены в диапазон [0..1].
- **Преобразование изображений**: каждое изображение 28×28 преобразовано в вектор длины 784 (для SVM и базового MLP).
- **Разбиение на обучающую и валидационную выборки**: 80% данных для обучения, 20% для теста (валидации).

Выводы

- Благодаря нормализации пикселей все значения признаков приводятся к единому масштабу, что особенно важно для методов, чувствительных к диапазону данных (например, SVM).
- Разделение на обучающую и валидационную части позволяет корректно оценивать качество моделей, избегая переобучения на всей выборке MNIST.

2. Классификация с помощью SVM

Описание этапа

1. **Масштабирование признаков**: используется `StandardScaler` из библиотеки scikit-learn для приведения каждого признака к среднему 0 и стандартному отклонению 1.

2. **PCA** (Principal Component Analysis): оставляем 98% дисперсии, что уменьшает размерность и отсеивает шумовые компоненты.

3. **Grid Search**: подбираем гиперпараметры `C` и `gamma` для SVM (ядро RBF). Для каждой комбинации проводится кросс-валидация (cv=3).

4. **Обучение**: по результатам Grid Search выбираем лучшую конфигурацию.

5. **Оценка на валидационной выборке**: получаем метрики точности, полноты и F1-score, а также строим матрицу ошибок.

Выводы

- Лучшая конфигурация: C=10, gamma=0.001 (при оставшихся 98% главных компонент).

- Точность на валидации достигла ~0.97, что говорит о том, что даже после снижения размерности SVM сохраняет высокую способность классифицировать цифры.

- Матрица ошибок показывает, что большая часть предсказаний лежит на главной диагонали, а ошибки распределены между близкими по написанию цифрами (например, «8» и «9»).

3. Классификация с помощью MLP

Описание этапа

1. **Архитектура сети**:
    - Flatten (28×28 → 784)
    - Dense(256) + Batch Normalization + Dropout(0.3)
    - Dense(128) + Batch Normalization + Dropout(0.3)
    - Dense(64) + Batch Normalization + Dropout(0.3)
    - Выходной слой: Dense(10, softmax)
2. **Оптимизатор**: Adam (learning_rate=0.001), далее автоматическое уменьшение скорости обучения (ReduceLROnPlateau).
3. **EarlyStopping**: останавливает обучение, если метрика (val_loss) не улучшается несколько эпох подряд.
4. **Обучение**: до 30 эпох, реальная остановка может произойти раньше из-за EarlyStopping.
5. **Оценка**: рассчитываем точность, строим матрицу ошибок и графики обучения (loss, accuracy).

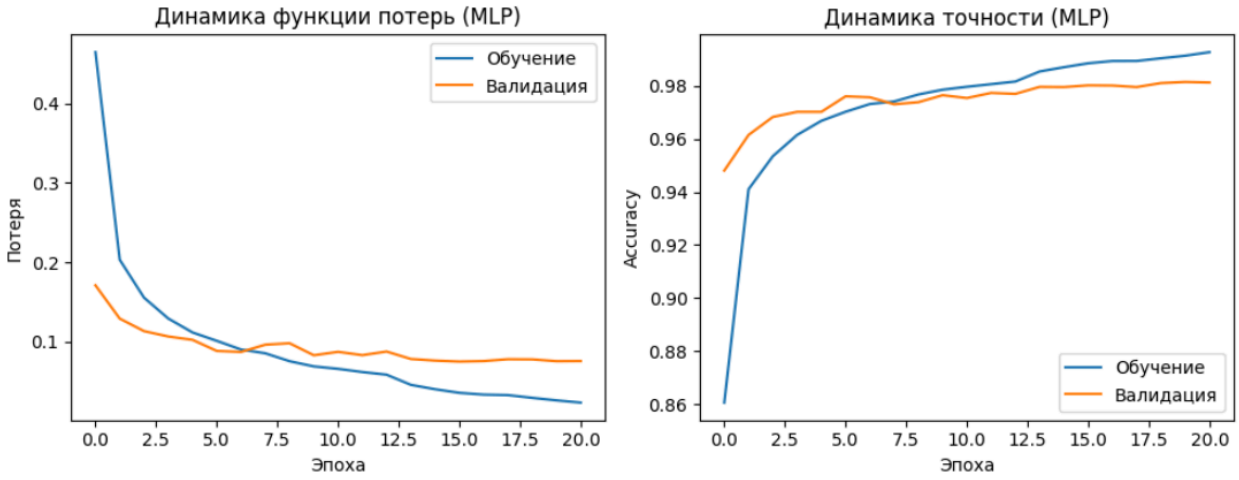
Выводы

- Точность на валидации ~0.98 — заметно улучшенный результат для MLP на MNIST (без сверточных слоёв).
- **Batch Normalization** помогает стабилизировать обучение, что видно по плавной динамике снижения функции потерь и роста точности.
- **Dropout** (0.3) уменьшает переобучение, сохраняя высокую точность на валидации.
- **EarlyStopping** и **ReduceLROnPlateau** позволили обучаться больше эпох, но при этом автоматически снижать скорость обучения и останавливать процесс при отсутствии улучшений, сохранив «лучшую» версию весов.

4. Анализ графиков обучения MLP

Описание этапа

- **График функции потерь (loss)**:
    - Линия «Обучение» (синяя) убывает от ~0.46 до ~0.03, показывая, что сеть продолжает подстраивать веса под обучающую выборку.
    - Линия «Валидация» (оранжевая) также плавно убывает, без резкого скачка вверх, что указывает на отсутствие серьёзного переобучения.
- **График точности (accuracy)**:
    - «Обучение» растёт почти до ~0.99.
    - «Валидация» достигает ~0.98 и сохраняется на этом уровне, небольшие колебания при обучении сглаживаются за счёт уменьшения learning rate.



Выводы

- Хорошее соотношение точности на обучающей и валидационной выборках (0.99 vs 0.98) говорит о том, что модель обобщает полученные знания, а не просто «запоминает» обучающие данные.
- Периодическое снижение learning rate (ReduceLROnPlateau) позволяет модели «тонко настроиться» и не застревать в локальных минимумах.

5. Итоговое сравнение моделей

Метрика	SVM (PCA + GridSearch)	MLP (BatchNorm + Dropout)
Accuracy	~0.97	~0.98
Recall	~0.97	~0.98
F1-score	~0.97	~0.98

Выводы

- **SVM**:
    - Достиг точности ~0.97.
    - Показал высокое качество на MNIST даже после уменьшения размерности (PCA).
    - Требует значительного времени на Grid Search (особенно при больших данных), но даёт интерпретируемые результаты.
- **MLP**:
    - Добился точности ~0.98 за счёт глубокого обучения с Batch Normalization, Dropout и адаптивной скоростью обучения.
    - Сеть более гибкая: при желании можно наращивать число слоёв, изменять размеры слоёв и т. д.
    - Потребовала меньше итераций подбора гиперпараметров (хотя тоже нужно подбирать оптимальную архитектуру).

Заключение

Обе модели продемонстрировали высокую точность (97–98%), что соответствует современному уровню решений на MNIST без использования сверточных сетей. При этом:

- **SVM** отлично работает при правильном подборе гиперпараметров и сокращении размерности (PCA).
- **MLP** при добавлении Batch Normalization и продвинутых техник обучения (EarlyStopping, ReduceLROnPlateau) получает более высокую точность.

Таким образом, в рамках ограничений «только SVM и MLP» обе модели являются жизнеспособными решениями для классификации изображений рукописных цифр, а предложенные улучшения (PCA, Grid Search, Batch Normalization, Dropout, адаптивная скорость обучения) позволяют повысить метрики до уровня ~97–98%.