Архитектуры нейронных сетей для прогнозирования:

Рекуррентные нейронные сети (RNN):

RNN предназначены для работы с последовательными данными, такими как временные ряды и текст.

Они имеют внутреннее состояние, которое позволяет им учитывать предыдущие значения входных данных.

Примеры: Simple RNN, LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit).

Сверточные нейронные сети (CNN):

CNN обычно ассоциируются с обработкой изображений, но они также могут быть применены к одномерным данным, таким как временные ряды.

Они извлекают признаки из локальных областей входных данных, что может быть полезным при анализе временных рядов с разными шаблонами.

Пример: 1D CNN.

Feedforward Neural Networks (FNN):

Это классические многослойные нейронные сети без внутренних обратных связей.

Используются в различных задачах прогнозирования, особенно в случаях, когда нет явной последовательности в данных.

Это один из двух широких типов искусственных нейронных сетей, характеризующийся направлением потока информации между ее слоями. Его поток является однонаправленным, что означает, что информация в модели течет только в одном направлении — вперед — от входных узлов через скрытые узлы (если таковые имеются) и к выходным узлам без каких-либо циклов или петель, в отличие от рекуррентных нейронных сетей, которые имеют двунаправленный поток. Современные сети прямого распространения обучаются с использованием метода обратного распространения ошибки и в просторечии называются «ванильными» нейронными сетями.

Transformer-based архитектуры:

Transformer-модели, которые изначально были разработаны для обработки текста, такие как BERT и GPT, также могут быть адаптированы для задач прогнозирования.

Они позволяют моделям учитывать долгосрочные зависимости в данных и обладают хорошей способностью к обобщению.

Способы обучения нейронных сетей для прогнозирования:

Случайное начальное обучение (Random Initialization):

Нейронные сети инициализируются случайными весами, и затем обучаются на тренировочных данных с использованием алгоритма обратного распространения ошибки.

Предобучение (Pretraining):

В некоторых случаях модели могут быть предварительно обучены на большом объеме данных и затем дообучены на специфической задаче прогнозирования.

Функции потерь (Loss Functions):

Выбор подходящей функции потерь зависит от типа задачи. Например, для регрессии часто используется среднеквадратичная ошибка (MSE), а для классификации – кросс-энтропия.

Оптимизация (Optimization):

Для обучения нейронных сетей используются оптимизационные алгоритмы, такие как стохастический градиентный спуск (SGD), Adam, RMSprop и другие.

Регуляризация (Regularization):

Для предотвращения переобучения моделей используется регуляризация, включая L1 и L2 регуляризацию и методы отсева (dropout).

Кросс-валидация (Cross-Validation):

Для оценки производительности модели и настройки гиперпараметров часто используется кросс-валидация.

Обратная связь и дообучение (Feedback and Fine-tuning):

Модели могут регулярно обновляться с использованием новых данных и обратной связи от реального мира.

Прогнозирование нелинейных процессов с использованием искусственных нейронных сетей (ИНС):

Подготовка данных:

Соберите и подготовьте данные для анализа. Удостоверьтесь, что данные корректны, заполните пропуски и проведите предварительный анализ данных (EDA) для выявления основных характеристик данных.

Выбор архитектуры ИНС:

Выберите подходящую архитектуру ИНС в зависимости от характера данных и задачи. Например, для последовательных данных, таких как временные ряды, рекуррентные нейронные сети (RNN) могут быть подходящим выбором. Для структурированных данных могут использоваться сверточные нейронные сети (CNN) или многослойные персептроны (FNN).

Подготовка обучающего и тестового наборов данных:

Разделите данные на обучающий и тестовый наборы для оценки производительности модели. Обычно используется метод отложенной выборки, кросс-валидация или временной раздел данных для временных рядов.

Создание модели ИНС:

Определите архитектуру ИНС, включая число слоев, количество нейронов в каждом слое и функции активации.

Выберите подходящую функцию потерь и оптимизатор для вашей задачи.

Обучение модели:

Обучите ИНС на обучающем наборе данных с использованием выбранной функции потерь и оптимизатора. Мониторьте процесс обучения и валидации, чтобы избежать переобучения.

Оценка производительности:

Используйте тестовый набор данных для оценки производительности модели. Вычислите метрики, такие как среднеквадратичная ошибка (MSE), коэффициент детерминации (R^2) или другие, в зависимости от задачи.

Настройка и оптимизация:

В случае необходимости проведите настройку гиперпараметров модели, таких как скорость обучения, количество эпох и структуру сети.

Прогнозирование:

После успешного обучения модели используйте ее для прогнозирования будущих значений на основе новых данных.

Мониторинг и обновление:

Регулярно мониторьте производительность модели в реальном времени и обновляйте ее при необходимости с учетом изменений в данных или бизнес-требованиях.