**Тема 3.8 Обзор задачи генерации**

**План**

1. Понятие «порождающие сети»
2. Примеры порождающих подходов
3. Вариационные автоэнкодеры
4. Генеративно-состязательные сети
5. Примеры подходов GAN

**Понятие «порождающие сети»**

Все рассмотренные до этого подходы, следует отнести к так называемы **дискременативным подходам**. Однако, возможен и другой путь. Это путь порождения результатов. То есть результат работы архитектуры не является на прямую решением, принятым на основе выделенных признаков во входных данных. Вместо этого сеть учится сама генерировать признаки на основе входного воздействия. Например, на основе случайного числа на входе. Тогда обучение работы сети сводится к сравнению результатов генерации с некоторыми эталонами. Эталоны имеют некоторые примеры тех признаков, которые учится генерировать сеть. Однако, примеры не обязаны покрывать все распределение значений признака, которое выучит сеть. Этот подход называется порождающим.

Порождающие подходы могу быть обучены любым способом, как задача с учителем или без него. В литературе предложен ряд подходов к решению задач порождающих сетей.

**Примеры порождающих подходов**

Среди порождающих подходов можно выделить два наиболее популярных:

* + Вариационные автоэнкодеры (Variational Autoencoders, VAE)
  + Генеративно-состязательные сети (Generative-Adversarial Network, GAN)

Эти и другие подходы могут комбинироваться.

**Вариационные автоэнкодеры**

**Вариационные автоэнкодеры (Variational Autoencoders, VAE)** — это автоэнкодеры, которые учатся отображать объекты в непрерывное скрытое пространство и, затем, воссоздавать их из него. Обычный автоэнкодер пытаемся восстановить информацию из скрытого пространства. Вариационный автоэнкодер пытаемся аппроксимировать скрытое пространство. Часто суть VAE в кодировании информации в виде нормального распределения.

**Генеративно-состязательные сети**

Другим гораздо более популярным подходом для генерации являются Порождающие состязательные сети. Это класс моделей, который обучается порождать по средствам сравнения генерируемых примеров и некоторых реальных данных. Подход содержит две части: **Сеть генератор** - порождает объекты в пространстве данных, и **Сеть дискриминатор**, которая учится отличать порожденные генератором объекты от настоящих примеров из обучающей выборки. В литературе предложено весьма много принципов организации генеративного-состязательных сетей.

**Примеры подходов GAN**

**Классический GAN** обучается без учителя. Есть примеры полуконтролируемого обучения. Например, **условный или Conditional GAN** – когда обучение дискреминатора и генератора проводятся с меткой класса. **Информационный InfoGAN** – когда генератор получает метку класса, а дискреминатор должен выучить эту метку самостоятельно.

В некоторых случаях может быть решена задача генерации из одних данных других. То есть перенос изображений или **Image 2 Image translation**. Примером такого подхода является **CycleGan** эта архитектура имеет структуру из энкодера – декодера – нового энкодера и нового декодера цель этой структуры сначала генерировать новое изображение, а затем из него обратно старое изображение.

**Резюме**

Генеративные подходы представляют сегодня достаточно быстро развивающуюся область в компьютерном зрении. Сегодня в литературе предложено много успешных реализаций порождающего подхода как связанных с ГАН и ВАЕ, так и вне их.