

## Contents

<b>1</b>	<b>stable-diffusion</b>	<b>2</b>
1.1	场景分析 . . . . .	2
1.1.1	应用场景 . . . . .	2
1.1.2	demo . . . . .	2
1.1.3	推理硬件需求 . . . . .	2
1.2	算子分析 . . . . .	2
1.2.1	原理 . . . . .	2
1.2.2	reference code . . . . .	3
1.2.3	算子类型 . . . . .	3
1.2.4	算子频度 . . . . .	3
1.3	网络分析 . . . . .	3
1.3.1	结构 . . . . .	3
1.3.2	参数 . . . . .	3
1.3.3	ifm sram . . . . .	3
1.3.4	带宽 . . . . .	3
1.4	部署 . . . . .	3
1.4.1	模型 . . . . .	3
1.4.2	框架 . . . . .	4
1.4.3	量化 . . . . .	4
1.4.4	硬件 . . . . .	4
1.4.5	PPA . . . . .	4
1.5	benchmark . . . . .	4
1.6	reference . . . . .	4
<b>2</b>	<b>swin transformer</b>	<b>4</b>
2.1	场景分析 . . . . .	4
2.1.1	应用场景 . . . . .	5
2.1.2	demo . . . . .	5
2.1.3	推理硬件需求 . . . . .	5
2.2	算子分析 . . . . .	5
2.2.1	原理 . . . . .	5
2.2.2	reference code . . . . .	5
2.2.3	算子类型 . . . . .	5
2.2.4	算子频度 . . . . .	5
2.3	网络分析 . . . . .	5
2.3.1	结构 . . . . .	5
2.3.2	参数 . . . . .	6
2.3.3	ifm sram . . . . .	6
2.3.4	带宽 . . . . .	6
2.4	部署 . . . . .	6
2.4.1	模型 . . . . .	6
2.4.2	框架 . . . . .	6
2.4.3	量化 . . . . .	6
2.4.4	硬件 . . . . .	6

2.4.5	PPA	6
2.5	benchmark	6
2.6	reference	6
<b>3</b>	<b>GPT-2</b>	<b>7</b>
3.1	场景分析	7
3.1.1	应用场景	7
3.1.2	demo	7
3.1.3	推理硬件需求	7
3.2	算子分析	7
3.3	网络分析	7
3.4	部署	7

## 1 stable-diffusion

- [官网](#)
- [论文](#)
- [Github地址](#)
- [模型v1-2](#) [模型v1-4](#)

### 1.1 场景分析

用户层使用模式，最好能用开源的网站试试，了解输入状况，场景状态

#### 1.1.1 应用场景

1. [AiDraw](#)
2. [civitai](#)
3. [数字绘画](#)
4. [openart.ai](#)

#### 1.1.2 demo

- [本地部署和配置](#)
  - install
  - usage

#### 1.1.3 推理硬件需求

### 1.2 算子分析

确认算子种类-频度

#### 1.2.1 原理

1. [The Illustrated Stable Diffusion](#)

2. [译-The Illustrated Stable Diffusion](#)
3. [文生图模型之Stable Diffusion](#)
4. [Stable Diffusion 原理介绍与源码分析](#)
5. [How does Stable Diffusion work?](#)

### 1.2.2 reference code

- [stable-diffusion-tensorflow](#)
- [stable-diffusion-pytorch](#)
- Diffusion Transformer

### 1.2.3 算子类型

### 1.2.4 算子频度

## 1.3 网络分析

确认权重参数大小，网络结构，带宽-sram考量

### 1.3.1 结构

- CLIP
- Encoder
- Decoder
- Diffusion
  - UNet
  - TimeEmbedding

### 1.3.2 参数

1. [Stable Diffusion模型运算量分析](#)
2. [onnx-tool](#)

### 1.3.3 ifm sram

### 1.3.4 带宽

## 1.4 部署

量化细节，数据类型...

### 1.4.1 模型

1. [Stable Diffusion Models](#)
2. [Stable Diffusion的模型量化](#)

#### 1.4.2 框架

- pytorch
- tensorflow

#### 1.4.3 量化

#### 1.4.4 硬件

- CUDA
- cpu
- tensor engine

#### 1.4.5 PPA

### 1.5 benchmark

### 1.6 reference

1. [Open-Assistant](#)
  2. [stable diffusion安装与简单评测](#)
  3. [35张图，直观理解Stable Diffusion](#)
  4. [stability](#)
  5. [denoising-diffusion-pytorch](#)
  6. [ControlNet](#)
  7. [sd-webui-controlnet](#)
  8. [AI数字绘画 stable-diffusion 保姆级教程](#)
  9. [stable-diffusion-webui-colab](#)
  10. [Stable Diffusion 原理介绍与源码分析](#)
  11. [使用Stable Diffusion UI手册](#)
  12. [Stable Diffusion的十个脑洞](#)
  13. [Colossal-AI](#)
- 

## 2 swin transformer

- [官网](#)
- [论文](#)
- [Github地址](#)
- [模型v1-2 模型v1-4](#)

### 2.1 场景分析

用户层使用模式，最好能用开源的网站试试，了解输入状况，场景状态

### 2.1.1 应用场景

1. [AiDraw](#)
2. [civitai](#)
3. [数字绘画](#)
4. [openart.ai](#)

### 2.1.2 demo

- [本地部署和配置](#)
  - install
  - usage

### 2.1.3 推理硬件需求

## 2.2 算子分析

确认算子种类-频度

### 2.2.1 原理

1. [The Illustrated Stable Diffusion](#)
2. [译-The Illustrated Stable Diffusion](#)
3. [文生图模型之Stable Diffusion](#)
4. [Stable Diffusion 原理介绍与源码分析](#)
5. [How does Stable Diffusion work?](#)

### 2.2.2 reference code

- [stable-diffusion-tensorflow](#)
- [stable-diffusion-pytorch](#)
- [Diffusion Transformer](#)

### 2.2.3 算子类型

### 2.2.4 算子频度

## 2.3 网络分析

确认权重参数大小，网络结构，带宽-sram考量

### 2.3.1 结构

- CLIP
- Encoder
- Decoder
- Diffusion

- UNet
- TimeEmbedding

### 2.3.2 参数

1. [Stable Diffusion模型运算量分析](#)
2. [onnx-tool](#)

### 2.3.3 ifm sram

### 2.3.4 带宽

## 2.4 部署

量化细节，数据类型...

### 2.4.1 模型

1. [Stable Diffusion Models](#)
2. [Stable Diffusion的模型量化](#)

### 2.4.2 框架

- pytorch
- tensorflow

### 2.4.3 量化

### 2.4.4 硬件

- CUDA
- cpu
- tensor engine

### 2.4.5 PPA

## 2.5 benchmark

## 2.6 reference

1. [Open-Assistant](#)
2. [stable diffusion安装与简单评测](#)
3. [35张图，直观理解Stable Diffusion](#)
4. [stability](#)
5. [denoising-diffusion-pytorch](#)
6. [ControlNet](#)
7. [sd-webui-controlnet](#)
8. [AI数字绘画 stable-diffusion 保姆级教程](#)
9. [stable-diffusion-webui-colab](#)

10. [Stable Diffusion 原理介绍与源码分析](#)
  11. [使用Stable Diffusion UI手册](#)
  12. [Stable Diffusion的十个脑洞](#)
  13. [Colossal-AI](#)
- 

## 3 GPT-2

### 3.1 场景分析

用户层使用模式，最好能用开源的网站试试，了解输入状况，场景状态

#### 3.1.1 应用场景

#### 3.1.2 demo

#### 3.1.3 推理硬件需求

### 3.2 算子分析

确认算子种类-频度

### 3.3 网络分析

确认权重参数大小，网络结构，带宽-sram考量

### 3.4 部署

量化细节，数据类型...

- 框架
  - 硬件
-