Explanation for Medical Text Embeddings & Neural Networks in PulseQuery AI – Yogesh Kumar Gopal (8996403)

Assignment Overview

Explain how neural networks generate medical text embeddings and use them as independent variables in the PulseQuery AI system.

Part 1: What Are Medical Text Embeddings?

Definition:

Medical text embeddings are **numbers that represent the meaning** of medical text. Instead of just looking at words, the computer converts medical sentences into **384 numbers** that capture what the medical text actually means.

Example:

- **Medical Text:** "Patient has chest pain and shortness of breath"
- **Embedding:** [0.23, -0.15, 0.87, 0.45, ...] (384 numbers total)
- **Why Useful:** The computer can now "understand" this means heart-related symptoms

Part 2: The Neural Network (MedEmbed)

What is MedEmbed?

MedEmbed is a **neural network** that was trained on millions of medical texts to understand medical language. https://huggingface.co/abhinand/MedEmbed-base-v0.1

Key Facts:

- **Name:** MedEmbed-base-v0.1
- **Type:** Transformer neural network (like ChatGPT, but for medical text)
- **Output:** 384 numbers for each medical sentence
- **Training:** Learned from real medical documents and patient records

How It Works:

1. **Input:** "67-year-old patient with diabetes"
2. **Processing:** Neural network analyzes the text
3. **Output:** 384 numbers that represent the medical meaning

Part 3: Code Implementation

Step 1: Loading the Neural Network

```
# Load the medical neural network
tokenizer = AutoTokenizer.from_pretrained("abhinand/MedEmbed-base-v0.1")
model = AutoModel.from_pretrained("abhinand/MedEmbed-base-v0.1")
```

Step 2: Converting Text to Numbers

```
def convert_medical_text_to_numbers(medical_text):
    # Prepare the text for the neural network
    inputs = tokenizer(medical_text, return_tensors="pt")

    # Run through neural network
    outputs = model(**inputs)

    # Get 384 numbers that represent the meaning
    embedding = outputs.last_hidden_state.mean(dim=1)

    return embedding  # Returns 384 numbers
```

Step 3: Using the Numbers

```
# Example usage
text1 = "Patient with heart attack"
text2 = "Patient with myocardial infarction"

embedding1 = convert_medical_text_to_numbers(text1)  # 384 numbers
embedding2 = convert_medical_text_to_numbers(text2)  # 384 numbers
```

```
# These embeddings will be very similar because they mean the same thing!
```

Part 4: How Embeddings Work as Independent Variables

In Simple Terms:

Independent variables are **inputs** that help predict something. In PulseQuery AI, the 384 embedding numbers are inputs that help predict energy savings.

Math:

```
Energy Efficiency = Function(384 embedding numbers + other factors)
```

How This Works:

- **Smart Understanding:** The 384 numbers capture medical meaning
- **Pattern Recognition:** The system learns which types of medical text can be compressed more
- **Prediction:** Based on the numbers, it predicts how much energy can be saved

Part 5: Real Example in PulseQuery AI

Original Medical Text:

"Patient Maria Rodriguez, 67 years old Hispanic female presenting to ED with chest pain and shortness of breath for the past 2 hours. Medical history includes hypertension and diabetes."

What Happens:

1. **Neural Network Input:** The text goes into MedEmbed
2. **Embedding Output:** 384 numbers that represent the medical meaning
3. **Analysis:** System understands this is about heart problems
4. **Optimization:** Safely compress to "Maria Rodriguez, 67yo F, c/o CP & SOB x 2hrs, h/o HTN & DM"
5. **Result:** 45% fewer words, same medical meaning, energy saved!

Part 6: Why This Matters

For Healthcare:

- **Energy Savings:** 30% less computer power needed
- **Cost Reduction:** Lower electricity bills for hospitals
- **Environmental Impact:** Less CO_2 emissions from computers
- **Safety:** Medical meaning is preserved

For AI:

- **Smart Compression:** Knows what can be shortened safely
- **Medical Accuracy:** Understands medical context
- **Real-time Processing:** Works in under 500 milliseconds